



UNIVERSIDAD DE LA RIOJA

TRABAJO FIN DE ESTUDIOS

Título

Control de un péndulo invertido rotatorio con hardware de bajo coste

Autor/es

FERNANDO CHICOTE NAVARIDAS

Director/es

JAVIER RICO AZAGRA y MONTSERRAT GIL MARTÍNEZ ,

Facultad

Escuela Técnica Superior de Ingeniería Industrial

Titulación

Grado en Ingeniería Electrónica Industrial y Automática

Departamento

INGENIERÍA ELÉCTRICA

Curso académico

2017-18



Control de un péndulo invertido rotatorio con hardware de bajo coste, de
FERNANDO CHICOTE NAVARIDAS
(publicada por la Universidad de La Rioja) se difunde bajo una Licencia Creative
Commons Reconocimiento-NoComercial-SinObraDerivada 3.0 Unported.
Permisos que vayan más allá de lo cubierto por esta licencia pueden solicitarse a los
titulares del copyright.



**UNIVERSIDAD
DE LA RIOJA**

ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA INDUSTRIAL

TRABAJO DE FIN DE GRADO

**TITULACIÓN: Grado en
Ingeniería Electrónica Industrial y Automática**

CURSO: 2017/2018

CONVOCATORIA: MARZO

TÍTULO:

**Control de un péndulo invertido rotatorio con hardware de
bajo coste**

AUTOR: Fernando Chicote Navaridas

DIRECTOR/ES: Javier Rico Azagra
Montserrat Gil Martínez

DEPARTAMENTO: Ingeniería Eléctrica

“CONTROL DE UN PÉNDULO INVERTIDO ROTATORIO CON HARDWARE DE BAJO COSTE”



Peticionario:	Universidad de La Rioja
Informantes:	Fernando Chicote Navaridas Estudiante de Ingeniería Electrónica Industrial y Automática Universidad de La Rioja
Lugar:	Logroño
Fecha:	lunes, 5 de marzo de 2018

RESUMEN

El presente trabajo fin de grado presenta la construcción, el estudio y el control de un péndulo invertido rotatorio, más conocido como péndulo de Furuta. En este estudio se describe la metodología empleada para modelar y controlar dicho péndulo. El péndulo de Furuta es un tipo de péndulo invertido construido a partir de dos brazos rotatorios acoplados en sus extremos. El brazo que gira en el plano horizontal es accionado por un motor de corriente continua. El movimiento en este primer brazo es trasladado al brazo vertical sobre el que se efectúa el control de posición.

Para el desarrollo del trabajo se ha construido un péndulo invertido rotatorio empleando materiales de bajo coste. En concreto, se utiliza la plataforma *LEGO MINDSTORMS EV3*, un motor de corriente continua *LEGO Power Functions XL* y dos encoders *GlideWheel-M Encoders*. Gracias a estos componentes puede obtenerse una plataforma sobre la que validar los resultados con un coste reducido.

En la primera parte del trabajo, se realiza un modelo dinámico del péndulo invertido, que es implementado en *Matlab-Simulink* para su simulación. Para obtener las ecuaciones que describen el comportamiento del sistema se utilizan las ecuaciones diferenciales del movimiento de *Euler-Lagrange*. Cabe destacar que el modelo obtenido presenta carácter no lineal.

Además, una de las características principales del sistema es su naturaleza sub-actuado, ya que dispone de dos salidas (ángulo del motor y ángulo del brazo vertical) a gobernar con una única acción de control (tensión de alimentación en bornes del motor). Véase como el ángulo en el motor, pese a no ser importante, debe controlarse para evitar que el sistema se equilibre con rotación constante. Los factores anteriores complican el control del sistema, siendo este un reto desde el punto de vista de la *Ingeniería de Control*.

Una vez modelado el comportamiento del sistema, se desarrolla un modelo lineal de pequeña señal en torno al punto de equilibrio sobre el que se desea gobernar el péndulo. Dicho modelo es expresado mediante ecuaciones matriciales en el espacio de estados. Se ha optado este modo de representación del comportamiento del sistema por considerarse más adecuado para el gobierno de un sistema sub-actuado.

RESUMEN

El control del sistema se efectúa empleando el método de control por realimentación de variables de estado. La finalidad del algoritmo de control es mantener el brazo vertical del péndulo en posición vertical hacia arriba, con oscilaciones mínimas y empleando incrementos en el ángulo del rotor del motor de corriente continua de amplitud mínima.

Para la obtención de las ganancias de realimentación de estado se utiliza la metodología de control óptimo (*LQR*). El control *LQR* permite, dado un sistema representado en el espacio de estados, encontrar las ganancias de realimentación óptimas según función de coste que penaliza las desviaciones de las variables de estado y de la señal de control.

En la segunda parte del trabajo se implementan los controladores sobre el péndulo invertido rotatorio que se ha construido. Para ello, se realiza una identificación y posterior validación de los parámetros que intervienen en el modelo matemático. Entre estos parámetros destacan las masas e inercias de los componentes y las constantes que determinan el comportamiento del motor de corriente continua. Una vez obtenidos los nuevos parámetros del sistema, se reajustan las ganancias de realimentación y se implementa el lazo de control en el sistema real.

Para ello es necesario desarrollar las herramientas de software en *Matlab* y *Simulink*. Con *Simulink* se desarrolla el sistema de control a cargar en el micro controlador del LEGO EV3, que lee los sensores, calcula la ley de control y la manda al actuador en tiempo real.

Además, utilizando también *Simulink*, se desarrolla una interface para el usuario. Esta interface establece una comunicación con el péndulo y monitoriza los datos de movimiento y actuación del mismo. Esta interface permite al usuario mandar comandos al péndulo para: arrancar o parar la actuación del motor, resetear los encoders o modificar el valor de las ganancias de realimentación. Con todos estos elementos se obtiene un conjunto software que permite controlar en tiempo real el estado del sistema, y además, permitiendo almacenar los datos generados durante los experimentos para su posterior análisis.

Palabras clave:

Péndulo invertido rotatorio, péndulo de Furuta, espacio de estados, control por realimentación de variables de estado y LEGO Mindstorms EV3.

ABSTRACT

This final degree project presents the construction, study and control of a rotary inverted pendulum, known as Furuta pendulum. In this study we describe the methodology used to model and control this pendulum. The Furuta Pendulum is a type of inverted pendulum built from two rotating arms coupled at their ends. The arm that rotates in the horizontal plane is driven by a direct current (DC) motor. The movement in this first arm is transferred to the vertical arm on which the position control is carried out.

For the development of the project a rotary inverted pendulum has been built using low cost materials. Specifically, the LEGO MINDSTORMS EV3 platform, a LEGO Power Functions XL DC motor and two GlideWheel-M Encoders are used. Thanks to these components, a platform can be built to validate the results with a reduced cost.

Firstly, a dynamic model of the rotary inverted pendulum is made, which is implemented in *Matlab-Simulink* for its simulation. To obtain the equations that describe the system behaviour, differential equations of the Euler-Lagrange movement are used. It should be noted that the final model has a non-linear character.

In addition, one of the main characteristics of the system is its sub-actuated nature: it has two outputs (the motor angle and the pendulum angle) to be regulated by a single control action (supply voltage at the motor terminals). Let us note that the motor angle, despite not being important, must be controlled to prevent the system from balancing with constant speed rotation. This factor makes a challenge the control of the system from the Control Engineering point of view.

Once the system behaviour is modelled, a small signal linear model is developed around the equilibrium point at which the pendulum is being controlled. This model is expressed with state space equations. This system representation has been considered as the most appropriated for the control of a sub-acted system.

The control of the system is performed using full-state feedback. The purpose of the control algorithm is to keep the vertical arm of the pendulum upright, with minimal oscillations and using increments of minimum amplitude in the rotor angle of the DC motor.

The linear quadratic regulator (LQR) methodology is used to obtain the full-state feedback gains. This control law finds the optimal full-state feedback gains according to a cost function that penalizes the deviations of the state variables and the control signal.

RESUMEN

Secondly, the controllers are implemented on the rotary inverted pendulum that has been built. To do this, an identification and a validation of the parameters that take part in the mathematical model is performed. These parameters include the masses and inertias of the components and the constants that determine the behaviour of the DC motor. Once the new system parameters have been obtained, the feedback gains are readjusted and the control loop is implemented in the real system.

For this it is necessary to develop the software tools in *Matlab* and *Simulink*. *Simulink* develops the control system to be deployed to the LEGO EV3 microcontroller, which reads the sensors, calculates the control law and sends it to the actuator in real time.

In addition, using *Simulink* also, an interface for the user is developed. This interface establishes a communication with the pendulum and monitors the movement and performance data of the pendulum. This interface allows the user to send commands to the pendulum to: start or stop the motor, reset the encoders or modify the value of the feedback gains. With all these elements a software set is obtained that allows to control in real time the system state, and also, to store the data generated during the experiments for its later analysis.

Keywords:

Rotary inverted pendulum, Furuta pendulum, state space representation, full-state feedback and LEGO Mindstorms EV3.



AGRADECIMIENTOS

A toda mi familia, en especial a mis padres y mi hermana, que siempre han estado apoyándome. Gracias a Javier Rico y Montse Gil, ya que sin ayuda no podría haber realizado este trabajo.

Por último, no puedo olvidarme de toda la gente que he conocido durante estos cuatro largos años, amigos, compañeros, profesores, que sin duda forman parte de todo esto.

“CONTROL DE UN PÉNDULO INVERTIDO ROTATORIO CON HARDWARE DE BAJO COSTE”



DOCUMENTO Nº 1 ÍNDICE GENERAL

Peticionario:	Universidad de La Rioja
Informantes:	Fernando Chicote Navaridas Estudiante de Ingeniería Electrónica Industrial y Automática Universidad de La Rioja

ÍNDICE GENERAL

MEMORIA	13
1. Introducción	13
Motivación	13
Antecedentes	14
Objeto del trabajo.....	15
Objetivos	15
2. Referencias	17
Bibliografía web	17
Libros bibliográficos	17
Artículos bibliográficos	17
3. Definiciones y abreviaturas	18
4. Modelo matemático del péndulo rotatorio invertido	19
Cinemática del sistema.....	20
Análisis energético del sistema	24
Ecuaciones del movimiento	27
Modelado del motor de corriente continua	31
Espacio de estados.....	32
Linealización	33
Puntos de equilibrio	34
Linealización alrededor del punto de equilibrio	35
5. Modelo del péndulo rotario invertido en Simulink	39
6. Diseño del controlador.....	47
Introducción	47
Regulación mediante realimentación de variables de estado	47
Controlabilidad y observabilidad	49
Cálculo de las ganancias de realimentación	51
Implementación LQR	54
Optimización de la ley de control	56
7. Control del péndulo rotatorio invertido real	64
Comportamiento del péndulo rotario invertido real.....	73
8. Conclusiones	81

ÍNDICE GENERAL

ANEXO 1. MONTAJE DE LA MAQUETA	85
ANEXO 2. SOPORTE LEGO MINDSTORMS EV3 SIMULINK	97
1. Primeros pasos con LEGO MINDSTORMS EV3 Hardware	98
Primer paso	98
Segundo paso	98
2. Librería de bloques LEGO MINDSTORMS EV3	99
Bloque Encoder	100
Bloque Motor	101
Bloque UDP Send	102
Bloque UDP Receive	103
3. Comunicación con LEGO MINDSTORMS EV3 Hardware	104
ANEXO 3. MOTOR DE CORRIENTE CONTINUA	105
1. Modelo matemático de un motor de corriente continua	105
2. Obtención experimental de la planta	108
3. Características del motor	109
PLIEGO DE CONDICIONES	116
1. Introducción al pliego de condiciones	116
2. Condiciones Generales	116
3. Condiciones Administrativas	117
4. Normativa y Reglamentación	118
Reglamento relacionado a productos electrónicos	118
Normativa relacionada con materiales y equipos	118
5. Condiciones facultativas	119
Dirección	119
Libro de órdenes	119
Modificaciones	119
6. Condiciones de materiales y equipos	120
Condiciones técnicas de los materiales	120
7. Condiciones económicas	121
Errores en el proyecto	121
Liquidación	121
8. Disposición final	121
MEDICIONES	124
PRESUPUESTO	127
1. Cuadro de precios	127
2. Presupuesto	128
3. Resumen del presupuesto	129

ÍNDICE DE FIGURAS

Ilustración 1. Representación del péndulo de furuta en el espacio	19
Ilustración 2. Descomposición del brazo horizontal del péndulo	20
Ilustración 3. Descomposición del brazo vertical del péndulo	21
Ilustración 4. Descomposición del péndulo sobre los ejes coordenados	21
Ilustración 5. Modelo del Péndulo Rotatorio Invertido en Simulink.....	39
Ilustración 6. Subsistema enmascarado del modelo del péndulo en Simulink.....	40
Ilustración 7. Subsistema Motor Lego XL	41
Ilustración 8. Parámetros del modelo del péndulo enmascarado.....	42
Ilustración 9. Editor de la máscara.....	43
Ilustración 10. Gráficas del movimiento del péndulo	45
Ilustración 11. Configuración del bloque Scope.....	45
Ilustración 12. Sistema de control en lazo cerrado para regulación	48
Ilustración 13. Implementación del controlador LQR en Simulink	56
Ilustración 14. Gráfica de la respuesta del ángulo del motor.....	59
Ilustración 15. Gráfica de la respuesta de la velocidad angular del motor	59
Ilustración 16. Gráfica de la respuesta de la velocidad angular del motor (5 segundos)	60
Ilustración 17. Gráfica de la respuesta del ángulo del péndulo	60
Ilustración 18. Gráfica de la respuesta del ángulo del péndulo (2 segundos).....	61
Ilustración 19. Gráfica de la respuesta de la velocidad angular del péndulo	61
Ilustración 20. Gráfica de la respuesta de la velocidad angular (2 segundos)	62
Ilustración 21. Gráfica de la acción de control	62
Ilustración 22. Gráfica de la acción de control (2 segundos).....	63
Ilustración 23. Gráfica de la acción de control (0,5 segundos).....	63
Ilustración 24. Modelo de Simulink ejecutado por el LEGO EV3.....	65
Ilustración 25. Subsistema Sensores.....	65
Ilustración 26. Subsistema Sensores -> Reset.....	66
Ilustración 27. Subsistema Sensores -> Lectura encoder Motor.....	66
Ilustración 28. Subsistema Sensores -> Leer encoder Pendulo.....	67
Ilustración 29. Subsistema Controlador LQR.....	68
Ilustración 30. Subsistema Controlador LQR ->Comunicación (Recibir).....	68
Ilustración 31. Subsistema Modo seguro	69
Ilustración 32. Subsistema Modo seguro -> Condición ángulos.....	69
Ilustración 33. Subsistema Comunicación (Enviar)	70
Ilustración 34. Subsistema Motor.....	70
Ilustración 35. Modelo de Simulink ejecutado en el ordenador	71
Ilustración 36. Comunicación ordenador - LEGO EV3	71
Ilustración 37. Parámetros del modelo de Simulink.....	72
Ilustración 38. Señales del péndulo	73
Ilustración 39. Interface Simulink Data Inspector.....	74
Ilustración 40. Log Selected Signals	74
Ilustración 41. Experimento 1.....	76
Ilustración 42. Experimento 2.....	77
Ilustración 43. Experimento 3.....	78
Ilustración 44. Experimento 4.....	79
Ilustración 45. Experimento 5.....	80
Ilustración 46. Kit LEGO MINDSTORMS Education EV3	85
Ilustración 47. Set de Expansión EV3.....	85
Ilustración 48. Adaptador USB inalámbrico	86
Ilustración 49. Motor LEGO Power Functions XL	86
Ilustración 50. GlideWheel-M Encoders.....	86

ÍNDICE GENERAL

Ilustración 51. Parte superior de la estructura	87
Ilustración 52. Parte superior de la estructura	87
Ilustración 53. Escuadras para la parte superior de la estructura	88
Ilustración 54. Anclajes para el ladrillo EV3 y base de la estructura	88
Ilustración 55. Base de la estructura.....	89
Ilustración 56. Parte inferior de la estructura	89
Ilustración 57. Parte superior de la estructura	90
Ilustración 58. Montaje del motor.....	91
Ilustración 59. Montaje del motor.....	91
Ilustración 60. Motor integrado en la parte superior	92
Ilustración 61. Estructura total.....	93
Ilustración 62. Montaje encoder y brazo del péndulo	94
Ilustración 63. Montaje encoder y brazo del péndulo	94
Ilustración 64. Estructura del péndulo invertido rotatorio.....	95
Ilustración 65. Estructura del péndulo invertido rotatorio.....	96
Ilustración 66. Comprobación versión de firmware.....	98
Ilustración 67. Dirección IP del dispositivo.....	98
Ilustración 68. Librería LEGO MINDSTORMS EV3.....	99
Ilustración 69. Bloque Encoder	100
Ilustración 70. Bloque Motor	101
Ilustración 71. Bloque UDP Send	102
Ilustración 72. Bloque UDP Receive.....	103
Ilustración 73. Esquema de un motor CC.....	105
Ilustración 74. Gráfica de la respuesta del motor ante la señal de entrada	108
Ilustración 75. Gráfica del modelo del motor	108
Ilustración 76. Gráfica Par-Velocidad-Corriente	109
Ilustración 77. Gráfica Tensión-Velocidad	112
Ilustración 78. Gráfica Tensión-Velocidad	112

ÍNDICE DE TABLAS

Tabla 1. Parámetros del péndulo rotario invertido.....	44
Tabla 2. Parámetros del motor de corriente continua.....	44
Tabla 3. Medidas experimento motor	109
Tabla 4. Datos obtenidos del experimento	111



“CONTROL DE UN PÉNDULO INVERTIDO ROTATORIO CON HARDWARE DE BAJO COSTE”



DOCUMENTO Nº 2 MEMORIA

Peticionario:	Universidad de La Rioja
Informantes:	Fernando Chicote Navaridas Estudiante de Ingeniería Electrónica Industrial y Automática

MEMORIA

1. Introducción

Motivación

En la actualidad las investigaciones sobre robótica han impulsado en gran medida el desarrollo de lo que se conoce como *Ingeniería de Control* debido a la necesidad de automatización existente en todo lo relacionado a procesos industriales que involucran el uso de maquinaria y por tanto, riesgo para las personas. Es por ello que cosas tan comunes hoy en día como el transporte, las comunicaciones y en general los procesos de producción han tenido una evolución enorme, alcanzando niveles de seguridad y calidad impensados algunas décadas atrás.

La *Ingeniería de Control* es la disciplina de la ingeniería que aplica la teoría de control para diseñar, planificar y desarrollar sistemas con comportamientos deseados. La práctica requiere utilización de sensores y actuadores de entrada para hacer modificaciones en la respuesta de salida. La ingeniería de control se enfoca principalmente en la implementación de sistemas de control que han sido diseñados a partir de modelos matemáticos del sistema que se quiere controlar.

Se ocupó desde sus orígenes de la automatización y del control automático de sistemas complejos, sin intervención humana directa. Campos como el control de procesos, control de sistemas electromecánicos, supervisión y ajuste de controladores y otros donde se aplican teorías y técnicas entre las que podemos destacar: Control óptimo, control predictivo, control robusto y control no lineal entre otros, todo ello con trabajos y aplicaciones muy diversas (investigación básica, investigación aplicada, militares, industriales, comerciales, etc.), las cuáles han hecho de la ingeniería de control una materia científica y tecnológica imprescindible hoy en día.

Conforme las plantas modernas con muchas entradas y salidas se vuelven más y más complejas, la descripción de un sistema de control moderno requiere una gran cantidad de ecuaciones. La teoría de control clásica, que trata de los sistemas con una entrada y una salida, pierde su potencialidad cuando se trabaja con sistemas con entradas y salidas múltiples.

Hacia 1960, debido a la disponibilidad de las computadoras digitales fue posible el análisis en el dominio del tiempo de sistemas complejos. La teoría de control moderna, basada en el análisis en el dominio del tiempo y la síntesis a partir de variables de estados, se ha desarrollado para manejar la creciente complejidad de las plantas modernas y los requisitos cada vez más exigentes sobre precisión peso y coste en aplicaciones militares, espaciales e industriales.

Antecedentes

La tendencia moderna en los sistemas de ingeniería es hacia una mayor complejidad, debido sobre todo a que se requieren tareas más complejas y buena precisión. Los sistemas complejos pueden tener múltiples entradas y múltiples salidas y pueden ser variantes en el tiempo. Debido a la necesidad de cumplir requisitos cada vez más exigentes en el comportamiento de los sistemas de control, el aumento en la complejidad del sistema y el fácil acceso a las computadoras a gran escala, la teoría moderna de control, que es una nueva aproximación al análisis y diseño de los sistemas de control complejo, se ha desarrollado desde 1960. Esta nueva aproximación se basa en el concepto de estado. El concepto de estado por sí mismo no es nuevo, puesto que ha existido durante bastante tiempo en el campo de la dinámica clásica y en otros campos.

La tendencia de control moderna contrasta con la teoría de control convencional en que su formulación es aplicable a sistemas de múltiples-entradas, múltiples-salidas, que pueden ser lineales o no lineales, invariantes en el tiempo o variables en el tiempo, mientras que la teoría convencional sólo es aplicable a sistemas de una entrada-una salida invariantes en el tiempo. Además, la teoría de control moderna es esencialmente una aproximación en el dominio temporal, mientras que la teoría de control convencional es una aproximación en el dominio de la frecuencia compleja.

Durante muchos años la investigación de la *Ingeniería de Control* se ha enfocado en el desarrollo y la comprensión de sistemas físicos que posean características no lineales dentro de su modelado matemático y en consecuencia un desafío para las técnicas clásicas de control, es por esta razón que los equipos conocidos como péndulos han sido elementos con gran acogida en el campo de la investigación, ya que son sistemas que poseen grandes similitudes con los modelos de equipos comúnmente utilizados en el mundo laboral, creándose así nuevos prototipos y esquemas de control que mejoren y/o superen lo que se ha hecho hasta el momento.

Desde la década de los 70's se empezó a trabajar en el sistema físico conocido como el péndulo invertido, el cual consta de características estáticas y dinámicas especiales que lo hacen atractivo para el desarrollo de estudios académicos y aplicaciones industriales. Dado su complejidad y limitantes físicas es quizás el sistema pendular sobre el cual se ha planteado más esquemas de control, y que por tanto ha permitido tener un conocimiento cualitativo y determinístico de la planta de gran valor y robustez, lográndose así resultados satisfactorios que siempre brindan la posibilidad de nuevos avances y abren la puerta para futuras investigaciones.

Una ventaja indudable de este tipo de planta es la capacidad y adaptabilidad a diferentes esquemas de control, sin embargo el péndulo invertido posee una limitante física inherente a sus componentes constructivas que restringen el movimiento del vehículo que lo sujeta, dado que la trayectoria que puede este recorrer está acotada por unos topes en los extremos que impiden su libre movimiento.

En el año de 1992 el Doctor Katsuhisa Furuta del Instituto Tecnológico de Tokio en Japón diseñó y construyó el péndulo invertido rotatorio y como su nombre indica se soporta sobre un eje que puede girar en sentido horario o contrario a este 360° , lo que en teoría significa que es independiente de la trayectoria, eliminando la limitante del modelo anterior.

Desde el momento en que se dio a conocer el sistema, conocido coloquialmente como péndulo de Furuta, se han realizado innumerables trabajos y artículos que tienen como objetivo específico aplicar una técnica o metodología para controlar y en lo posible garantizar la estabilidad del brazo (péndulo) en una región o zona deseada, conocida como punto de equilibrio.

Objeto del trabajo

El péndulo invertido rotatorio es también conocido como péndulo de Furuta y es un mecanismo sub-actuado con dos grados de libertad y constituido por dos eslabones o brazos y dos articulaciones rotacionales. El primer eslabón tiene la capacidad de girar en el plano horizontal. En uno de sus extremos se encuentra el actuador (motor), quien imprime sobre este la capacidad de movimiento. A su vez el extremo del primer eslabón se conecta al segundo eslabón, o péndulo en el extremo, restante mediante una articulación que puede girar libremente y permite su rotación sobre el plano vertical.

La característica por la cual es definido como un mecanismo sub-actuado es debido a que solo posee un actuador (motor) en el primer brazo y tiene la capacidad de moverse en dos planos diferentes y perpendiculares entre sí. En consecuencia posee más grados de libertad de movimiento que actuadores cumpliendo así con la definición de sistema sub-actuado.

Objetivos

De acuerdo con el objeto del TFG, existe un objetivo principal que consiste en diseñar un sistema de control para la estabilización de un péndulo invertido rotatorio en la posición vertical superior mediante la realimentación de variables de estado. Para conseguir el objetivo principal del trabajo, se desarrollará:

- El estudio y la realización de un modelo matemático del Péndulo de Furuta e implementación en *Simulink*.
- El diseño de un controlador para la estabilización del péndulo.
- El montaje de la maqueta con la plataforma *LEGO Mindstorms EV3*.
- El control del péndulo rotatorio invertido real.
- El diseño de una interface para enviar parámetros de control y monitorizar el péndulo rotatorio invertido desde un ordenador.

MEMORIA

Mediante mecánica lagrangiana se estudiarán las ecuaciones matemáticas que definen el movimiento del péndulo invertido rotatorio, para su posterior tratamiento e implementación en *Simulink*. Además se estudiarán los elementos que forman parte del péndulo real para poder obtener los parámetros que lo caracterizan.

Utilizando el modelo realizado se obtendrá un controlador para la estabilización del péndulo. Para ello, primero el modelo será linealizado en torno a un punto de equilibrio y estará definido en el espacio de estados. Después, mediante realimentación de variables de estado y utilizando el método LQR (Linear Quadratic Regulator) para obtención de las ganancias de realimentación se obtendrá un control óptimo para el sistema.

Se realizará el montaje del péndulo rotatorio invertido real mediante la plataforma *LEGO Mindstorms EV3*. Sobre esta maqueta se implementará el controlador diseñado sobre el modelo del péndulo.

Paralelamente, se va a diseñar una pequeña aplicación que permitirá obtener los datos de funcionamiento de la maqueta, además de modificar las ganancias de realimentación, iniciar o finalizar el movimiento del motor y realizar un reseteo de los sensores que forman parte del sistema.

2. Referencias

Bibliografía web

- <https://es.mathworks.com>
- <https://www.lego.com/es-es/mindstorms>
- <http://www.mindsensors.com>
- <http://www.philohome.com/motors/motorcomp.htm>
- http://www.dashhub.org/unlv/wiki/doku.php?id=lego_rip_initial

Libros bibliográficos

- Katsuhiko Ogata. Ingeniería de Control Moderna. Pearson Educación, 2003

Artículos bibliográficos

- Carlos Andrés Osorio Zúñiga. Diseño, construcción y control de un péndulo invertido rotacional utilizando técnicas lineales y no lineales. Tesis para optar al título de Magister en automatización industrial, Universidad Nacional de Colombia, Bogotá, 2009.
- Carlos Regalo Núñez. Control y Simulación del Péndulo de Furuta. Trabajo Fin de Grado, Universidad de Sevilla, 2016.
- César Vieira Rocha. Realização de um controlador utilizando Redes Neurais para o controle de estabilização do Pêndulo de Furuta. Proyecto final de curso, Universidade Federal do Amazonas, Manaus, 2010.
- Leidy Tatiana Poveda Galvis y José David Martínez Velasco. Diseño e implementación de un control LQR con la tarjeta Raspberry Pi. Trabajo de grado para optar por el título de Ingeniero de Control, Universidad Distrital Francisco José de Caldas, Bogotá, 2016.
- Luisa Fernanda Escobar Dávila y Óscar Danilo Montoya Giraldo. Control Híbrido de un péndulo de furuta. Proyecto de grado presentado como requisito para optar por el título de Ingeniero (a) Electricista, Universidad Tecnológica de Pereira, 2012.
- Magnus Gäfvert. Modelling the Furuta Pendulum. Department of Automatic Control. Lund Institute of Technology, 1998.

3. Definiciones y abreviaturas

- Espacio de estados: Modelo matemático de un sistema físico descrito mediante un conjunto de entradas, salidas y variables de estado relacionadas por ecuaciones diferenciales que se combinan en una ecuación diferencial matricial de primer orden.
- Lego Mindstorms EV3: Tercera generación de sets de robótica perteneciente a la línea de LEGO Mindstorms, y sucesor del kit NXT. Las siglas “EV” hacen referencia a las siglas en inglés de “evolution”, lo cual en español significa evolución en español; y el número “3”, representa la generación del set.
- LQR: Siglas de *Linear-Quadratic Regulator*, que hace referencia al método de control implementado en el péndulo rotatorio invertido.
- Matlab: Herramienta de software matemático que ofrece un entorno de desarrollo integrado (IDE) con un lenguaje de programación propio (lenguaje M).
- Mecánica lagrangiana: Reformulación de la mecánica clásica introducida por Joseph-Louis de Lagrange en 1788. En la mecánica lagrangiana, la trayectoria de un objeto es obtenida encontrando la trayectoria que minimiza la acción, que es la integral del Lagrangiano en el tiempo; siendo éste la energía cinética del objeto menos la energía potencial del mismo.
- Péndulo: Sistema físico que puede oscilar bajo la acción gravitatoria u otra característica física y que está configurado por una masa suspendida de un punto o de un eje horizontal fijos mediante un hilo, una varilla, u otro dispositivo que sirve para medir el tiempo.
- Péndulo invertido: Péndulo que tiene su centro de masa por encima de su punto de pivote.
- Péndulo rotatorio invertido: Más conocido como péndulo de Furuta, consiste en un brazo impulsado que gira en el plano horizontal y un péndulo unido a ese brazo que puede girar libremente en el plano vertical.
- PWM: Viene del inglés *Pulse Width Modulation*, cuyo significado es modulación por ancho de pulsos en español. Está referida a la acción de control que se le envía al motor.
- Simulink: Entorno de programación visual, que funciona sobre el entorno de programación Matlab.
- TFG: Siglas de Trabajo Fin de Grado.
- UDP: Viene del inglés *User Data Protocol*, es un protocolo del nivel de transporte basado en el intercambio de datagramas (Encapsulado de capa 4 o de Transporte del Modelo OSI). Permite el envío de datagramas a través de la red sin que se haya establecido previamente una conexión, ya que el propio datagrama incorpora suficiente información de direccionamiento en su cabecera.

4. Modelo matemático del péndulo rotatorio invertido

El sistema conocido como péndulo de Furuta se muestra en la Ilustración 1. Como se puede observar consta de dos cuerpos inerciales conectados. Un pilar central (eje del motor) con un momento de inercia J_m (inercia total equivalente en el origen, que incluye la inercia de los brazos y del motor eléctrico), rígidamente conectado al brazo horizontal de longitud l_a y masa homogéneamente distribuida m_a . En su extremo a se articula el péndulo con longitud l_p y masa homogéneamente distribuida m_p . Además se considera una masa de balance en el punto superior p del péndulo de magnitud M . El ángulo del péndulo θ se ha definido como cero en la posición vertical superior, y positivo, cuando el péndulo se mueve en la dirección de las manecillas del reloj. El ángulo del brazo horizontal ϕ se ha definido como positivo cuando el brazo se mueve en la dirección contraria a las manecillas del reloj.

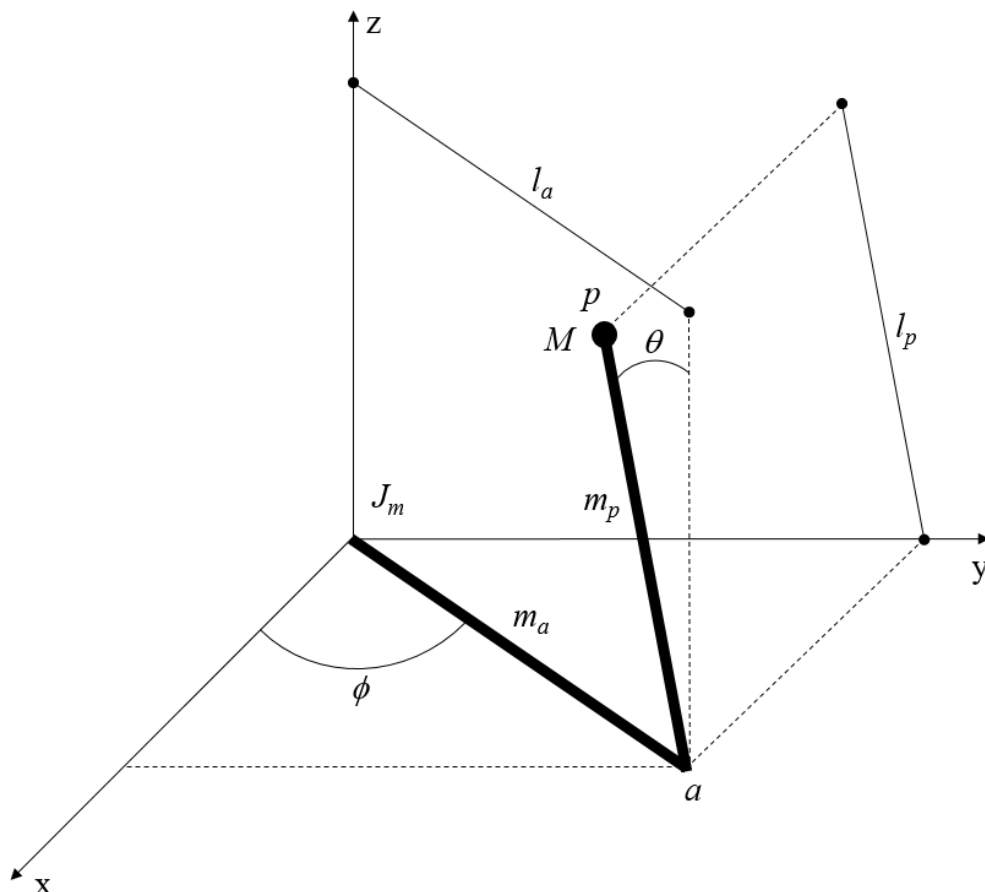


Ilustración 1. Representación del péndulo de furuta en el espacio

Cinemática del sistema

La posición en el espacio de un punto p sobre el péndulo puede ser descrita por el vector de posición que se muestra en la siguiente ecuación (4.1).

$$r(r_a, r_p) = (r_x(r_a, r_p), r_y(r_a, r_p), r_z(r_a, r_p))^T \quad (4.1)$$

Donde las variables r_a y r_p son las posiciones radiales de los brazos horizontal y vertical del péndulo respectivamente, las cuáles son medidas desde los centros de rotación de los cuerpos.

Realizando la descomposición sobre los ejes coordenados se obtienen las ecuaciones de posición del péndulo (4.2), (4.3) y (4.4).

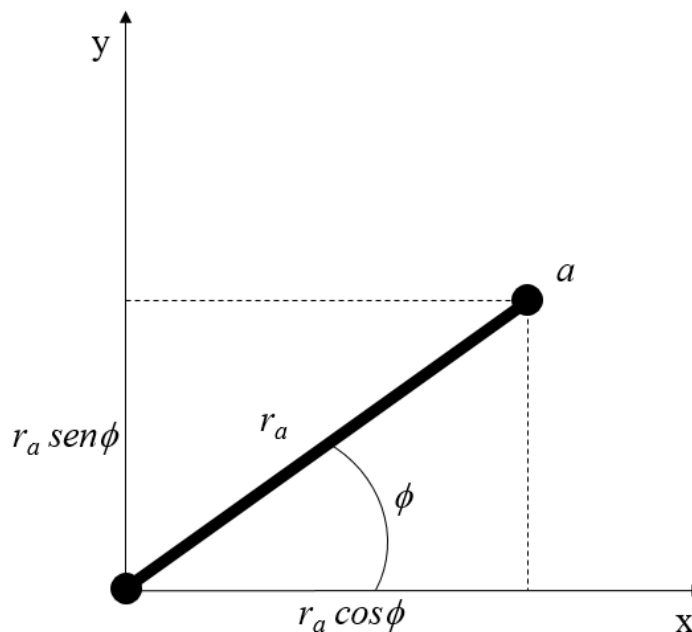


Ilustración 2. Descomposición del brazo horizontal del péndulo

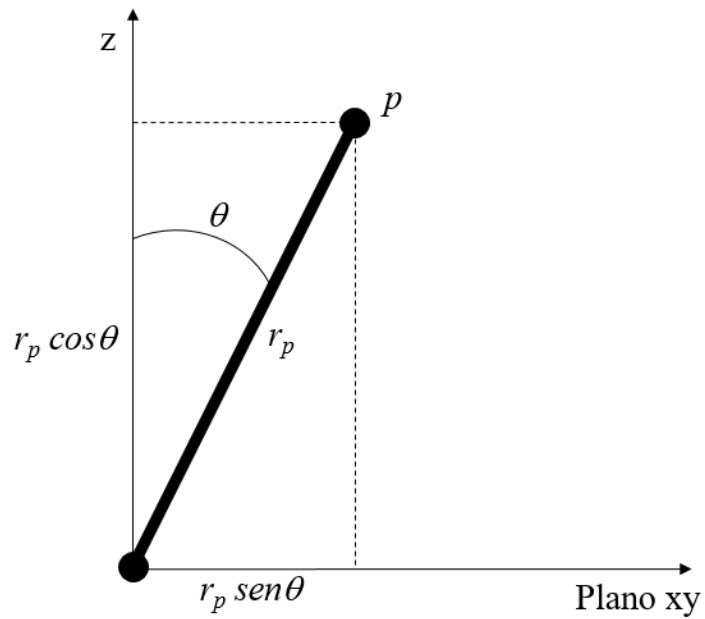


Ilustración 3. Descomposición del brazo vertical del péndulo

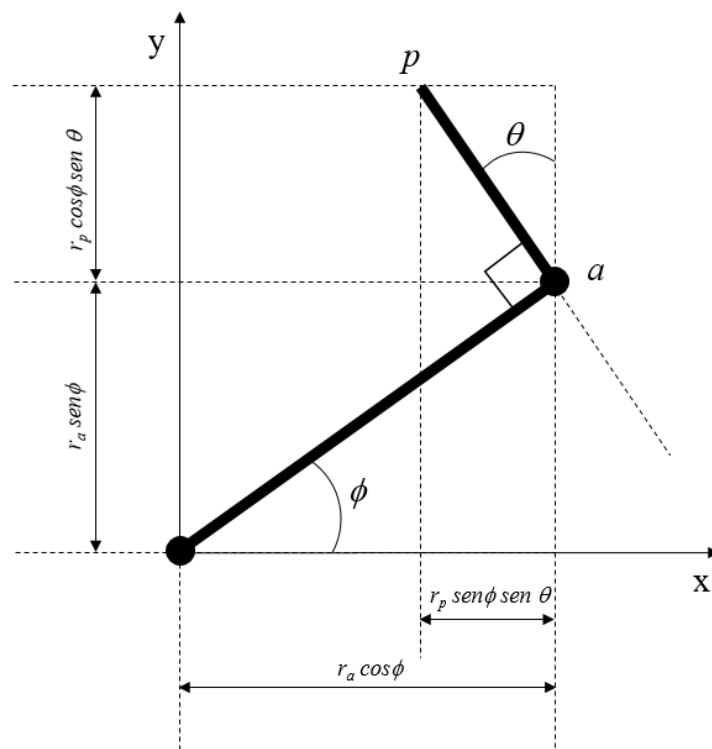


Ilustración 4. Descomposición del péndulo sobre los ejes coordenados

MEMORIA

$$r_x(r_a, r_p) = r_a \cos \phi - r_p \sin \phi \sin \theta \quad (4.2)$$

$$r_y(r_a, r_p) = r_a \sin \phi + r_p \cos \phi \sin \theta \quad (4.3)$$

$$r_z(r_a, r_p) = r_p \cos \theta \quad (4.4)$$

Tomando la derivada con respecto al tiempo de (4.1) se obtiene la ecuación vectorial de la velocidad (4.5).

$$v(r_a, r_p) = (v_x(r_a, r_p), v_y(r_a, r_p), v_z(r_a, r_p))^T \quad (4.5)$$

Tomando las derivadas con respecto del tiempo de las ecuaciones (4.2), (4.3) y (4.4) se obtienen las velocidades correspondientes al sistema en cada eje coordenado como se puede ver en las ecuaciones (4.6), (4.7) y (4.8).

$$v_x(r_a, r_p) = -r_a \sin \phi \dot{\phi} - r_p \cos \phi \sin \theta \dot{\phi} - r_p \sin \phi \cos \theta \dot{\theta} \quad (4.6)$$

$$v_y(r_a, r_p) = r_a \cos \phi \dot{\phi} - r_p \sin \phi \sin \theta \dot{\phi} + r_p \cos \phi \cos \theta \dot{\theta} \quad (4.7)$$

$$v_z(r_a, r_p) = -r_p \sin \theta \dot{\theta} \quad (4.8)$$

Teniendo en cuenta que para obtener la energía cinética es necesario conocer la magnitud de la velocidad total del sistema, en la siguiente ecuación (4.9) se muestra como calcular dicha magnitud para un sistema de tres componentes.

$$v^2(r_a, r_p) = v_x^2(r_a, r_p) + v_y^2(r_a, r_p) + v_z^2(r_a, r_p) \quad (4.9)$$

En consecuencia mediante la manipulación algebraica de las ecuaciones (4.6), (4.7) y (4.8), y efectuando las respectivas simplificaciones se obtiene la ecuación de velocidad (4.10).

$$\begin{aligned}v_x^2(r_a, r_p) &= r_a^2 \sin^2 \phi \dot{\phi}^2 + r_a r_p \sin \phi \cos \phi \sin \theta \dot{\phi}^2 + r_a r_p \sin^2 \phi \cos \theta \dot{\phi} \dot{\theta} \\&+ r_a r_p \sin \phi \cos \phi \sin \theta \dot{\phi}^2 + r_p^2 \cos^2 \phi \sin^2 \theta \dot{\phi}^2 + r_p^2 \sin \phi \cos \phi \sin \theta \cos \theta \dot{\phi} \dot{\theta} \\&+ r_a r_p \sin^2 \phi \cos \theta \dot{\phi} \dot{\theta} + r_p^2 \sin \phi \cos \phi \sin \theta \cos \theta \dot{\phi} \dot{\theta} + r_p^2 \sin^2 \phi \cos^2 \theta \dot{\phi}^2 \\v_y^2(r_a, r_p) &= r_a^2 \cos^2 \phi \dot{\phi}^2 - r_a r_p \sin \phi \cos \phi \sin \theta \dot{\phi}^2 + r_a r_p \cos^2 \phi \cos \theta \dot{\phi} \dot{\theta} \\&- r_a r_p \sin \phi \cos \phi \sin \theta \dot{\phi}^2 + r_p^2 \sin^2 \phi \sin^2 \theta \dot{\phi}^2 - r_p^2 \sin \phi \cos \phi \sin \theta \cos \theta \dot{\phi} \dot{\theta} \\&+ r_a r_p \cos^2 \phi \cos \theta \dot{\phi} \dot{\theta} - r_p^2 \sin \phi \cos \phi \sin \theta \cos \theta \dot{\phi} \dot{\theta} + r_p^2 \cos^2 \phi \cos^2 \theta \dot{\phi}^2 \\v_z^2(r_a, r_p) &= -r_p^2 \sin^2 \theta \dot{\theta}^2 \\v^2(r_a, r_p) &= (r_a^2 + r_p^2 \sin^2 \theta) \dot{\phi}^2 + 2r_a r_p \cos \theta \dot{\phi} \dot{\theta} + r_p^2 \dot{\theta}^2\end{aligned}\tag{4.10}$$

Análisis energético del sistema

Después de que se ha calculado la magnitud de la velocidad para el sistema del péndulo rotatorio invertido, es posible encontrar las expresiones de la energía total de la planta. La energía total del sistema quedará dividida en cuatro elementos diferenciados que forman el péndulo: energía del centro de rotación, energía del brazo horizontal, energía del brazo vertical y energía de la masa de balance. Las energías se denominarán U y K , como energía potencial gravitatoria y energía cinética respectivamente.

Las siguientes ecuaciones (4.11) y (4.12) representan las expresiones para calcular las energías presentes en el sistema del péndulo de Furuta:

$$K = \int \frac{1}{2} v^2 dm \quad (4.11)$$

$$U = g \int r_z dm \quad (4.12)$$

Donde dm es un diferencial de masa y g la constante de gravedad.

Los cálculos para calcular las energías se realizan de la siguiente forma:

Centro de rotación

La energía potencial gravitatoria (U) en el centro de rotación es cero, ya que se encuentra en el origen de coordenadas.

$$K_{CR} = \int \frac{1}{2} J_m \dot{\phi}^2 dm$$

$$K_{CR} = \frac{1}{2} J_m \dot{\phi}^2 \quad (4.13)$$

$$U_{CR} = 0 \quad (4.14)$$

Brazo horizontal

La energía potencial gravitatoria (U) en el brazo horizontal también es cero, debido a que el brazo está fijado al centro de rotación que como se ha mencionado anteriormente, se encuentra en el origen de coordenadas.

$$K_{BH} = \int_0^{l_a} v^2(s, 0) \frac{m_a}{l_a} ds$$

$$K_{BH} = \frac{1}{2} \frac{m_a}{l_a} \left(\int_0^{l_a} r_a^2 \dot{\phi}^2 ds + \int_0^{l_a} r_p^2 \sin^2 \theta \dot{\phi}^2 ds + \int_0^{l_a} 2r_a r_p \cos \theta \dot{\phi} \dot{\theta} ds + \int_0^{l_a} r_p^2 \dot{\theta}^2 ds \right)$$

$$K_{BH} = \frac{1}{2} \frac{m_a}{l_a} \dot{\phi}^2 \left[\frac{r_a^3}{3} \right]_0^{l_a}$$

$$K_{BH} = \frac{1}{2} \left(\frac{1}{3} m_a l_a \dot{\phi}^2 \right) \quad (4.15)$$

$$U_{BH} = 0 \quad (4.16)$$

Brazo vertical

$$K_{BV} = \int_0^{l_p} v^2(l_a, s) \frac{m_p}{l_p} ds$$

$$K_{BV} = \frac{1}{2} \frac{m_p}{l_p} \left(\int_0^{l_p} r_a^2 \dot{\phi}^2 ds + \int_0^{l_p} r_p^2 \sin^2 \theta \dot{\phi}^2 ds + \int_0^{l_p} 2r_a r_p \cos \theta \dot{\phi} \dot{\theta} ds + \int_0^{l_p} r_p^2 \dot{\theta}^2 ds \right)$$

$$K_{BH} = \frac{1}{2} \frac{m_p}{l_p} \left(l_a^2 \dot{\phi}^2 \left[r_p \right]_0^{l_p} + \sin^2 \theta \dot{\phi}^2 \left[\frac{r_p^3}{3} \right]_0^{l_p} + 2l_a \cos \theta \dot{\phi} \dot{\theta} \left[\frac{r_p^2}{2} \right]_0^{l_p} + \dot{\theta}^2 \left[\frac{r_p^3}{3} \right]_0^{l_p} \right)$$

$$K_{BH} = \frac{1}{2} m_p l_a^2 \dot{\phi}^2 + \frac{1}{6} m_p l_p^2 \sin^2 \theta \dot{\phi}^2 + \frac{1}{2} m_p l_a l_p \cos \theta \dot{\phi} \dot{\theta} + \frac{1}{6} m_p l_p^2 \dot{\theta}^2$$

$$K_{BH} = \frac{1}{2} m_p \left(l_a^2 + \frac{1}{3} l_p^2 \sin^2 \theta \right) \dot{\phi}^2 + \frac{1}{2} m_p l_a l_p \cos \theta \dot{\phi} \dot{\theta} + \frac{1}{6} m_p l_p^2 \dot{\theta}^2 \quad (4.17)$$

$$U_{BH} = g \int_0^{l_p} r_z(l_a, s) \frac{m_p}{l_p} ds$$

$$U_{BH} = g \frac{m_p}{l_p} \int_0^{l_p} r_p \cos \theta ds$$

$$U_{BH} = g \frac{m_p}{l_p} \cos \theta \left[\frac{r_p^2}{2} \right]_0^{l_p}$$

$$U_{BH} = \frac{1}{2} g m_p l_p \cos \theta \quad (4.18)$$

Masa de balance

$$K_{MB} = \int v^2(l_a, l_p) M ds$$

$$K_{MB} = \frac{1}{2} M \left(\int r_a^2 \dot{\phi}^2 ds + \int r_p^2 \sin^2 \theta \dot{\phi}^2 ds + \int 2 r_a r_p \cos \theta \dot{\phi} \dot{\theta} ds + \int r_p^2 \dot{\theta}^2 ds \right)$$

$$K_{MB} = \frac{1}{2} M \left(l_a^2 \dot{\phi}^2 + l_p^2 \sin^2 \theta \dot{\phi}^2 + 2 l_a l_p \cos \theta \dot{\phi} \dot{\theta} + l_p^2 \dot{\theta}^2 \right)$$

$$K_{MB} = \frac{1}{2} M \left(l_a^2 + l_p^2 \sin^2 \theta \right) \dot{\phi}^2 + M l_a l_p \cos \theta \dot{\phi} \dot{\theta} + \frac{1}{2} M l_p^2 \dot{\theta}^2 \quad (4.19)$$

$$U_{MB} = g \int r_z M ds$$

$$U_{MB} = M g \int r_p \cos \theta ds$$

$$U_{MB} = M g l_p \cos \theta \quad (4.20)$$

MEMORIA

Una vez calculadas todas las energías (cinética y potencial) del sistema, es preciso el cálculo total de las energías, que es igual a la suma de todas las energías parciales del sistema. Para el cálculo de la energía cinética total se usa $K=K_{CR}+ K_{BH}+ K_{BV}+ K_{MB}$ y para la energía potencial total se usa $U=U_{CR}+ U_{BH}+ U_{BV}+ U_{MB}$, como se puede ver en las ecuaciones(4.21) y (4.22).

$$K = \frac{1}{2} \left[J_m \dot{\phi}^2 + \frac{1}{3} m_a l_a \dot{\phi}^2 + m_p \left(l_a^2 + \frac{1}{3} l_p^2 \sin^2 \theta \right) \dot{\phi}^2 + m_p l_a l_p \cos \theta \dot{\phi} \dot{\theta} \right] + \frac{1}{2} \left[\frac{1}{3} m_p l_p^2 \dot{\theta}^2 + M \left(l_a^2 + l_p^2 \sin^2 \theta \right) \dot{\phi}^2 + 2 M l_a l_p \cos \theta \dot{\phi} \dot{\theta} + M l_p^2 \dot{\theta}^2 \right] \quad (4.21)$$

$$U = \frac{1}{2} g m_p l_p \cos \theta + M g l_p \cos \theta \quad (4.22)$$

Ecuaciones del movimiento

Después de encontrar las ecuaciones que definen las energías el sistema, es necesario hacer un estudio sobre el método de Lagrange, pues este método permite modelar el péndulo y encontrar las expresiones de movimiento en función de la entrada activa (par). Las ecuaciones (4.23) y (4.24) definen este método.

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{\phi}} \right) - \frac{\partial L}{\partial \phi} = \tau_{\phi} \quad (4.23)$$

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{\theta}} \right) - \frac{\partial L}{\partial \theta} = \tau_{\theta} \quad (4.24)$$

Donde L , conocido como Lagrangiano, representa la diferencia entre la energía cinética y potencial. En la ecuación (4.25) se obtiene el Lagrangiano.

$$L = K - U \quad (4.25)$$

$$\begin{aligned} L = & \frac{1}{2} J_m \dot{\phi}^2 + \frac{1}{6} m_a l_a^2 \dot{\phi}^2 + \frac{1}{2} m_p l_a^2 \dot{\phi}^2 + \frac{1}{6} m_p l_p^2 \sin^2 \theta \dot{\phi}^2 + \frac{1}{2} m_p l_a l_p \cos \theta \dot{\phi} \dot{\theta} + \\ & \frac{1}{6} m_p l_p^2 \dot{\theta}^2 + \frac{1}{2} M l_a^2 \dot{\phi}^2 + \frac{1}{2} M l_p^2 \sin^2 \theta \dot{\phi}^2 + M l_a l_p \cos \theta \dot{\phi} \dot{\theta} + \frac{1}{2} M l_p^2 \dot{\theta}^2 - \\ & \frac{1}{2} g m_p l_p \cos \theta - M g l_p \cos \theta \end{aligned} \quad (4.26)$$

Como se puede observar en las ecuaciones (4.23) y (4.24) es necesario encontrar los términos que las componen. Se realizan las derivadas parciales para luego sustituirlas en las ecuaciones de Lagrange.

Componente ϕ

En la ecuación del Lagrangiano (4.26) no existe ningún término que corresponda con la posición angular ϕ y por ello la derivada parcial con respecto a este término es cero.

$$\frac{\partial L}{\partial \phi} = 0 \quad (4.27)$$

Componente $\dot{\phi}$

Calculando la derivada parcial con respecto a la velocidad angular del brazo horizontal en la ecuación (4.26) se obtiene la ecuación (4.28).

$$\begin{aligned} \frac{\partial L}{\partial \dot{\phi}} = & J_m \dot{\phi} + \frac{1}{3} m_a l_a^2 \dot{\phi} + m_p l_a^2 \dot{\phi} + \frac{1}{3} m_p l_p^2 \sin^2 \theta \dot{\phi} + \frac{1}{2} m_p l_a l_p \cos \theta \dot{\theta} + \\ & M l_a^2 \dot{\phi} + M l_p^2 \sin^2 \theta \dot{\phi} + M l_a l_p \cos \theta \dot{\theta} \\ \frac{\partial L}{\partial \dot{\phi}} = & \left[J_m + \left(M + \frac{1}{3} m_a + m_p \right) l_a^2 + \left(M + \frac{1}{3} m_p \right) l_p^2 \sin^2 \theta \right] \dot{\phi} + \left(M + \frac{1}{2} m_p \right) l_a l_p \cos \theta \dot{\theta} \end{aligned} \quad (4.28)$$

Componente θ

Realizando la derivada parcial con respecto a la posición angular del brazo vertical en la ecuación (4.26) se obtiene la expresión (4.29).

$$\begin{aligned}\frac{\partial L}{\partial \theta} &= \frac{1}{3} m_p l_p^2 \sin \theta \cos \theta \dot{\phi}^2 - \frac{1}{2} m_p l_a l_p \sin \theta \dot{\phi} \dot{\theta} + M l_p^2 \sin \theta \cos \theta \dot{\phi}^2 - \\ &\quad M l_a l_p \sin \theta \dot{\phi} \dot{\theta} + \frac{1}{2} g m_p l_p \sin \theta + M g l_p \sin \theta \\ \frac{\partial L}{\partial \theta} &= \left(M + \frac{1}{3} m_p \right) l_p^2 \sin \theta \cos \theta \dot{\phi}^2 - \left(M + \frac{1}{2} m_p \right) l_a l_p \sin \theta \dot{\phi} \dot{\theta} + \left(M + \frac{1}{2} m_p \right) g l_p \sin \theta\end{aligned}\quad (4.29)$$

Componente $\dot{\theta}$

Al realizar la derivada parcial con respecto a la velocidad angular del brazo vertical en la ecuación (4.26) se obtiene la ecuación (4.30).

$$\begin{aligned}\frac{\partial L}{\partial \dot{\theta}} &= \frac{1}{2} m_p l_a l_p \cos \theta \dot{\phi} + \frac{1}{3} m_p l_p^2 \dot{\theta} + M l_a l_p \cos \theta \dot{\phi} + M l_p^2 \dot{\theta} \\ \frac{\partial L}{\partial \dot{\theta}} &= \left(M + \frac{1}{2} m_p \right) l_a l_p \cos \theta \dot{\phi} + \left(M + \frac{1}{3} m_p \right) l_p^2 \dot{\theta}\end{aligned}\quad (4.30)$$

Para simplificar las ecuaciones (4.27), (4.28), (4.29) y (4.30) además facilitar el cálculo de las derivadas temporales, se definen unos parámetros auxiliares en función de las constantes propias del sistema físico (4.31).

$$\begin{aligned}\alpha &= J_m + \left(M + \frac{1}{3} m_a + m_p \right) l_a^2 \\ \beta &= \left(M + \frac{1}{3} m_p \right) l_p^2 \\ \gamma &= \left(M + \frac{1}{2} m_p \right) l_a l_p \\ \delta &= \left(M + \frac{1}{2} m_p \right) g l_p\end{aligned}\quad (4.31)$$

MEMORIA

Al sustituir estas nuevas constantes en las ecuaciones (4.27), (4.28), (4.29) y (4.30), se obtienen (4.32), (4.33), (4.34) y (4.35).

$$\frac{\partial L}{\partial \phi} = 0 \quad (4.32)$$

$$\frac{\partial L}{\partial \dot{\phi}} = (\alpha + \beta \sin^2 \theta) \dot{\phi} + \gamma \cos \theta \dot{\theta} \quad (4.33)$$

$$\frac{\partial L}{\partial \theta} = \beta \sin \theta \cos \theta \dot{\phi}^2 - \gamma \sin \theta \dot{\phi} \dot{\theta} + \delta \sin \theta \quad (4.34)$$

$$\frac{\partial L}{\partial \dot{\theta}} = \gamma \cos \theta \dot{\phi} + \beta \dot{\theta} \quad (4.35)$$

Tomando la derivada con respecto del tiempo de la expresión (4.33), se llega a la ecuación (4.36).

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{\phi}} \right) = (\alpha + \beta \sin^2 \theta) \ddot{\phi} + \gamma \cos \theta \ddot{\theta} + 2\beta \sin \theta \cos \theta \dot{\phi} \dot{\theta} - \gamma \sin \theta \dot{\theta}^2 \quad (4.36)$$

Tomando la derivada con respecto del tiempo de la expresión (4.35), se llega a la ecuación (4.37).

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{\theta}} \right) = \gamma \cos \theta \ddot{\phi} + \beta \ddot{\theta} - \gamma \sin \theta \dot{\phi} \dot{\theta} \quad (4.37)$$

Finalmente al realizar la combinación lineal que se muestra en las ecuaciones (4.23) y (4.24), utilizando los resultados obtenidos en las ecuaciones (4.32), (4.34), (4.36) y (4.37) se llega a las ecuaciones (4.38) y (4.39) que describen la dinámica del sistema en términos del par de entrada.

$$(\alpha + \beta \sin^2 \theta) \ddot{\phi} + \gamma \cos \theta \ddot{\theta} + 2\beta \sin \theta \cos \theta \dot{\phi} \dot{\theta} - \gamma \sin \theta \dot{\theta}^2 = \tau_{\phi} \quad (4.38)$$

$$\gamma \cos \theta \ddot{\phi} + \beta \ddot{\theta} - \beta \sin \theta \cos \theta \dot{\phi}^2 - \delta \sin \theta = \tau_{\theta} \quad (4.39)$$

Modelado del motor de corriente continua

Teniendo en cuenta que el par de un motor de corriente continua, está relacionado con el voltaje aplicado en los terminales del motor, es necesario hallar la expresión matemática que representa el circuito eléctrico del motor en términos de las variables físicas de corriente, voltaje y velocidad de giro en el eje del motor, como se observa en la ecuación (4.40).

$$V(t) = Ri(t) + L \frac{di(t)}{dt} + K_e \dot{\phi}(t) \quad (4.40)$$

Donde R , L y K_e son constantes eléctricas del motor: Resistencia, autoinducción de los devanados y constante de fuerza contraelectromotriz.

En la ecuación (4.40), se puede despreciar el efecto inductivo por ser muy pequeño ($L \approx 0$). Considerando que el par está relacionado con la corriente de armadura a través de la constante de par K_t , se obtiene el modelo simplificado, presente en las ecuaciones (4.41) y (4.42).

$$i(t) = \frac{1}{R} V(t) - \frac{K_e}{R} \dot{\phi}(t) \quad (4.41)$$

$$\tau_\phi(t) = K_t i(t) \quad (4.42)$$

Sustituyendo (4.41) en (4.42), se obtiene la expresión (4.43) que relaciona el par con el voltaje aplicado en los terminales del motor.

$$\tau_\phi(t) = \frac{K_t}{R} V(t) - \frac{K_e K_t}{R} \dot{\phi}(t) \quad (4.43)$$

Espacio de estados

A partir del modelo matemático del péndulo de Furuta presente en las ecuaciones (4.38) y (4.39) e introduciendo las variables de estado $x_1 = \phi$, $x_2 = \dot{\phi}$, $x_3 = \theta$ y $x_4 = \dot{\theta}$, es posible reescribir el comportamiento de la planta como una serie de ecuaciones diferenciales de primer orden no lineales, como se muestra en las ecuaciones (4.44), (4.45), (4.46) y (4.47) que además representan el modelo de estados. Además se añade el factor disipativo de la fricción viscosa (b) para el grado de libertad ϕ .

$$\dot{x}_1 = x_2 \quad (4.44)$$

$$\begin{aligned} \dot{x}_2 = & \frac{\beta(\tau_\phi - bx_2) - \beta\gamma \cos^2 x_3 \sin x_3 x_2^2 - 2\beta^2 \cos x_3 \sin x_3 x_2 x_4}{\alpha\beta - \gamma^2 + (\beta^2 + \gamma^2) \sin^2 x_3} \\ & + \frac{\beta\gamma \sin x_3 x_4^2 - \gamma\delta \cos x_3 \sin x_3}{\alpha\beta - \gamma^2 + (\beta^2 + \gamma^2) \sin^2 x_3} \end{aligned} \quad (4.45)$$

$$\dot{x}_3 = x_4 \quad (4.46)$$

$$\begin{aligned} \dot{x}_4 = & \frac{\beta(\alpha + \beta \sin^2 x_3) \cos x_3 \sin x_3 x_2^2 + 2\beta\gamma \cos^2 x_3 \sin x_3 x_2 x_4}{\alpha\beta - \gamma^2 + (\beta^2 + \gamma^2) \sin^2 x_3} \\ & + \frac{-\gamma^2 \cos x_3 \sin x_3 x_4^2 + \delta(\alpha + \beta \sin^2 x_3) \sin x_3 - \gamma^2 \cos x_3 (\tau_\phi - bx_2)}{\alpha\beta - \gamma^2 + (\beta^2 + \gamma^2) \sin^2 x_3} \end{aligned} \quad (4.47)$$

Linealización

La linealización se utiliza para transformar un sistema no lineal en un sistema lineal equivalente que será válido para pequeños desplazamientos alrededor de un punto de equilibrio. Existen diferentes formas de abordar una solución para este problema. Expandir las ecuaciones del estado no lineal en series de Taylor alrededor de un punto de equilibrio es una de ellas.

Considerando una función $f(x)$, y suponiendo que \bar{x} es un punto donde $f(\bar{x}) = 0$. En este caso, el punto \bar{x} es llamado punto de equilibrio del sistema del sistema $\dot{x} = f(x)$, desde $x = 0$ (sistema en el origen) hasta $x = \bar{x}$ (sistema en el punto de equilibrio). La expansión en series de Taylor de $f(x)$ entorno al punto \bar{x} viene dada por (4.48) y (4.49).

$$f(x) = \sum_{n=0}^{\infty} \frac{1}{n!} \left. \frac{d^n f(x)}{dx^n} \right|_{x=\bar{x}} (x - \bar{x})^n \quad (4.48)$$

$$f(x) = f(\bar{x}) + \left. \frac{df(x)}{dx} \right|_{x=\bar{x}} (x - \bar{x}) + \frac{1}{2} \left. \frac{d^2 f(x)}{dx^2} \right|_{x=\bar{x}} (x - \bar{x})^2 + \frac{1}{6} \left. \frac{d^3 f(x)}{dx^3} \right|_{x=\bar{x}} (x - \bar{x})^3 \dots \quad (4.49)$$

Puede ser escrita como (4.50), despreciando los términos de orden superior a 1.

$$f(x) = f(\bar{x}) + \left. \frac{df(x)}{dx} \right|_{x=\bar{x}} (x - \bar{x}) \quad (4.50)$$

Para completar la linealización se toma $\delta x = (x - \bar{x})$.

Puntos de equilibrio

Para conocer los puntos de equilibrio de la planta, donde las derivadas se hacen cero, se igualan a cero las ecuaciones no lineales (4.44), (4.45), (4.46) y (4.47). Se obtiene el sistema de ecuaciones no lineales (4.51).

$$\begin{aligned}x_2 &= 0 \\ \cos x_3 \sin x_3 &= 0 \\ x_4 &= 0 \\ \cos x_3 \sin x_3 &= 0 \\ \delta(\alpha + \beta \sin^2 x_3) \sin x_3 &= 0\end{aligned}\tag{4.51}$$

Por lo tanto resolviendo el sistema de ecuaciones, los puntos de equilibrio son todos los puntos que cumplan $(x_1, 0, x_3, 0)$. Donde $x_1 \in \mathbb{R}$ y $x_3 = k\pi$, con $k \in \mathbb{Z}$.

El ángulo del péndulo θ se ha definido como cero en la posición vertical superior, por lo tanto el punto de equilibrio seleccionado, en torno al cual se van a expandir las ecuaciones en series de Taylor, es el punto $(0, 0, 0, 0)$.

Linealización alrededor del punto de equilibrio

Una vez conocidos los puntos de equilibrio se pueden expandir las ecuaciones (4.44), (4.45), (4.46) y (4.47) en series de Taylor en torno al punto de equilibrio elegido.

Siendo el conjunto de ecuaciones no lineales (4.52).

$$\dot{x} = f(x, \tau, t) \quad (4.52)$$

En donde x y u son vectores columna que contienen n variables de estado y p entradas al sistema, respectivamente; y f es una función vectorial no lineal de x , τ y t .

Sea un punto de operación detonado por $x_0(t)$, que corresponde a la entrada $\tau_0(t)$. Expandiendo la ecuación no lineal (4.52) en series de Taylor alrededor del punto de equilibrio (x_0, τ_0) y despreciando los términos de orden superior a 1, se llega a la expresión (4.53).

$$\dot{x}_i(t) = f_i(x_0, \tau_0) + \sum_{j=1}^n \left. \frac{\partial f_i(x, \tau)}{\partial x_j} \right|_{x_0, \tau_0} (x_j - x_{0j}) + \sum_{j=1}^p \left. \frac{\partial f_i(x, \tau)}{\partial \tau_j} \right|_{x_0, \tau_0} (\tau_j - \tau_{0j}) \quad (4.53)$$

En donde $i = 1, 2, 3, \dots, n$.

Se definen:

$$\delta x_j \cong x_j - x_{0j} \quad (4.54)$$

$$\delta \tau_j \cong \tau_j - \tau_{0j} \quad (4.55)$$

$$\delta \dot{x}_i \cong \dot{x}_i - \dot{x}_{0i} \quad (4.56)$$

En dónde $\dot{x}_{0i} = f_i(x_0, \tau_0)$.

Sustituyendo (4.54), (4.55) y (4.56) en la expresión (4.53), se obtiene la ecuación (4.57).

$$\delta \ddot{x}_i = f_i(x_0, \tau_0) + \sum_{j=1}^n \left. \frac{\partial f_i(x, \tau)}{\partial x_j} \right|_{x_0, \tau_0} \delta x_j + \sum_{j=1}^p \left. \frac{\partial f_i(x, \tau)}{\partial \tau_j} \right|_{x_0, \tau_0} \delta \tau_j \quad (4.57)$$

Aplicando la ecuación (4.57) para las ecuaciones (4.44), (4.45), (4.46) y (4.47).

$$\begin{aligned} \delta \ddot{x}_1 = & f_1(x_0, \tau_0) + \left. \frac{\partial f_1(x, \tau)}{\partial x_1} \right|_{x_0, \tau_0} \delta x_1 + \left. \frac{\partial f_1(x, \tau)}{\partial x_2} \right|_{x_0, \tau_0} \delta x_2 + \\ & \left. \frac{\partial f_1(x, \tau)}{\partial x_3} \right|_{x_0, \tau_0} \delta x_3 + \left. \frac{\partial f_1(x, \tau)}{\partial x_4} \right|_{x_0, \tau_0} \delta x_4 + \left. \frac{\partial f_1(x, \tau)}{\partial \tau_1} \right|_{x_0, \tau_0} \delta \tau_1 \end{aligned} \quad (4.58)$$

$$\begin{aligned} \delta \ddot{x}_2 = & f_2(x_0, \tau_0) + \left. \frac{\partial f_2(x, \tau)}{\partial x_1} \right|_{x_0, \tau_0} \delta x_1 + \left. \frac{\partial f_2(x, \tau)}{\partial x_2} \right|_{x_0, \tau_0} \delta x_2 + \\ & \left. \frac{\partial f_2(x, \tau)}{\partial x_3} \right|_{x_0, \tau_0} \delta x_3 + \left. \frac{\partial f_2(x, \tau)}{\partial x_4} \right|_{x_0, \tau_0} \delta x_4 + \left. \frac{\partial f_2(x, \tau)}{\partial \tau_1} \right|_{x_0, \tau_0} \delta \tau_1 \end{aligned} \quad (4.59)$$

$$\begin{aligned} \delta \ddot{x}_3 = & f_3(x_0, \tau_0) + \left. \frac{\partial f_3(x, \tau)}{\partial x_1} \right|_{x_0, \tau_0} \delta x_1 + \left. \frac{\partial f_3(x, \tau)}{\partial x_2} \right|_{x_0, \tau_0} \delta x_2 + \\ & \left. \frac{\partial f_3(x, \tau)}{\partial x_3} \right|_{x_0, \tau_0} \delta x_3 + \left. \frac{\partial f_3(x, \tau)}{\partial x_4} \right|_{x_0, \tau_0} \delta x_4 + \left. \frac{\partial f_3(x, \tau)}{\partial \tau_1} \right|_{x_0, \tau_0} \delta \tau_1 \end{aligned} \quad (4.60)$$

$$\begin{aligned} \delta \ddot{x}_4 = & f_4(x_0, \tau_0) + \left. \frac{\partial f_4(x, \tau)}{\partial x_1} \right|_{x_0, \tau_0} \delta x_1 + \left. \frac{\partial f_4(x, \tau)}{\partial x_2} \right|_{x_0, \tau_0} \delta x_2 + \\ & \left. \frac{\partial f_4(x, \tau)}{\partial x_3} \right|_{x_0, \tau_0} \delta x_3 + \left. \frac{\partial f_4(x, \tau)}{\partial x_4} \right|_{x_0, \tau_0} \delta x_4 + \left. \frac{\partial f_4(x, \tau)}{\partial \tau_1} \right|_{x_0, \tau_0} \delta \tau_1 \end{aligned} \quad (4.61)$$

Resolviendo (4.58), (4.59), (4.60) y (4.61).

$$\delta \ddot{x}_1 = \delta \ddot{x}_2 \quad (4.62)$$

$$\delta \ddot{x}_2 = \frac{-\beta b}{\alpha\beta - \gamma^2} \delta \ddot{x}_2 + \frac{-\delta\gamma}{\alpha\beta - \gamma^2} \delta \ddot{x}_3 + \frac{\beta}{\alpha\beta - \gamma^2} \delta \tau_1 \quad (4.63)$$

$$\delta \ddot{x}_3 = \delta \ddot{x}_4 \quad (4.64)$$

$$\delta \ddot{x}_4 = \frac{\gamma b}{\alpha\beta - \gamma^2} \delta \ddot{x}_2 + \frac{\alpha\delta}{\alpha\beta - \gamma^2} \delta \ddot{x}_3 + \frac{-\gamma}{\alpha\beta - \gamma^2} \delta \tau_1 \quad (4.65)$$

Las ecuaciones (4.62), (4.63), (4.64) y (4.65) se pueden reescribir de forma matricial como se muestra en la expresión (4.66).

$$\frac{d(\delta \dot{x})}{dt} = A \delta \dot{x} + B \delta \tau \quad (4.66)$$

La ecuación que relaciona la salida del sistema con las variables de estado y las entradas viene dada por la ecuación (4.67).

$$\delta y = C \delta \dot{x} + D \delta \tau \quad (4.67)$$

MEMORIA

Después de linealizar el modelo matemático en el punto de equilibrio (0, 0, 0, 0) vertical superior con las variables de estado que se definieron, se obtiene el modelo del sistema en espacio de estados (4.68).

$$\begin{pmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \end{pmatrix} = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & \frac{-\beta b}{\alpha\beta - \gamma^2} & \frac{-\delta\gamma}{\alpha\beta - \gamma^2} & 0 \\ 0 & 0 & 0 & 1 \\ 0 & \frac{\gamma b}{\alpha\beta - \gamma^2} & \frac{\alpha\delta}{\alpha\beta - \gamma^2} & 0 \end{pmatrix} \cdot \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} + \begin{pmatrix} 0 \\ \frac{\beta}{\alpha\beta - \gamma^2} \\ 0 \\ \frac{-\gamma}{\alpha\beta - \gamma^2} \end{pmatrix} \cdot (\tau_\phi) \quad (4.68)$$

$$\begin{pmatrix} y_1 \\ y_2 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \cdot \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \end{pmatrix} \cdot (\tau_\phi)$$

Es de notar que en la matriz C, las variables de interés, o sea las salidas, son $x_1 = \phi$ y $x_3 = \theta$.

En el anterior modelo matemático se considera el par aplicado al motor como entrada al sistema. En la ecuación (4.43), se relaciona el par del motor con el voltaje aplicado en los terminales del mismo. Utilizando el resultado de la expresión (4.43) en el modelo representado en (4.68), se obtiene (4.69).

$$\begin{pmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \end{pmatrix} = \begin{pmatrix} 0 & \frac{1}{-\beta(K_t^2 + Rb)} & \frac{0}{\alpha\beta - \gamma^2} & 0 \\ 0 & \frac{-\beta(K_t^2 + Rb)}{R(\alpha\beta - \gamma^2)} & \frac{-\delta\gamma}{\alpha\beta - \gamma^2} & 0 \\ 0 & 0 & 0 & 1 \\ 0 & \frac{\gamma(K_t^2 + Rb)}{R(\alpha\beta - \gamma^2)} & \frac{\alpha\delta}{\alpha\beta - \gamma^2} & 0 \end{pmatrix} \cdot \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} + \begin{pmatrix} 0 \\ \frac{\beta K_t}{R(\alpha\beta - \gamma^2)} \\ 0 \\ \frac{-\gamma K_t}{R(\alpha\beta - \gamma^2)} \end{pmatrix} \cdot (V) \quad (4.69)$$

$$\begin{pmatrix} y_1 \\ y_2 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \cdot \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \end{pmatrix} \cdot (V)$$

5. Modelo del péndulo rotario invertido en Simulink

Simulink es una herramienta muy potente y sencilla para resolver ecuaciones diferenciales y así asumir el comportamiento del movimiento. A continuación se va a explicar cómo se han introducido las ecuaciones diferenciales que definen el comportamiento del péndulo.

Utilizando las ecuaciones (4.44), (4.45), (4.46) y (4.47), que describen el comportamiento de la planta como una serie de ecuaciones diferenciales de primer orden no lineales, se puede obtener el modelo de simulación del péndulo.

La ecuación (4.44) define la velocidad angular del brazo horizontal ($\dot{\phi}$), la (4.45) hace referencia a la derivada de la ecuación anterior, la (4.46) define la velocidad angular del brazo vertical ($\dot{\theta}$) y por último la (4.47) representa la derivada de la expresión anterior.

El péndulo rotario invertido real implementado con la plataforma *LEGO Mindstorms EV3* está controlado por una señal PWM. Se trata de un modelo en el que se considera como entrada el voltaje entre los terminales del motor. Este modelo se puede observar en la Ilustración 5.

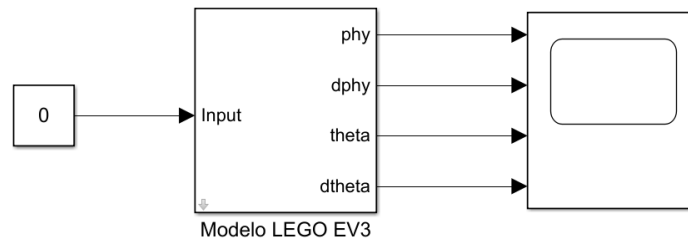


Ilustración 5. Modelo del Péndulo Rotatorio Invertido en Simulink

Crear un subsistema es útil principalmente para ahorrar espacio y poder ver de forma más clara el modelo que se ha creado, para ello solo es necesario seleccionar la parte del modelo que se quiere introducir dentro del subsistema, pulsar el botón derecho y seleccionar **Create Subsystem from Selection**.

Lo que se encuentra dentro del subsistema, para poder acceder a ello se pulsa el botón derecho sobre el subsistema se selecciona **Mask > Look Under Mask**.

En el subsistema que se puede apreciar en la Ilustración 6, se puede observar:

- La entrada al sistema es la tensión aplicada entre los bornes del motor (PWM).
- Diez constantes que definen las características reales del péndulo que son: Inercia del motor, masa del brazo horizontal, masa del brazo vertical, masa de balance, longitud del brazo horizontal, longitud del brazo vertical, gravedad, constantes eléctricas del motor y fricción viscosa.
- Un bloque *MATLAB Function*, es un bloque que permite introducir código de *Matlab* en un archivo *Simulink*, que define las ecuaciones diferenciales. Más adelante se muestra el código que contiene dicho bloque.
- Un integrador, en el que se definen las condiciones iniciales en velocidad y posición. Las condiciones iniciales son: $\phi = 0$, $\dot{\phi} = 0$, $\theta = 0.1$ y $\dot{\theta} = 0$. Este integrador calcula las variables de estado, ya que del anterior bloque únicamente de obtienen las derivadas de las mismas.
- Las cuatro salidas del subsistema de definen las cuatro variables de estado del péndulo: posición angular del motor, velocidad angular del motor, posición angular del péndulo y velocidad angular del péndulo.
- Un subsistema *Motor Lego XL*, que define la conversión PWM a voltios.

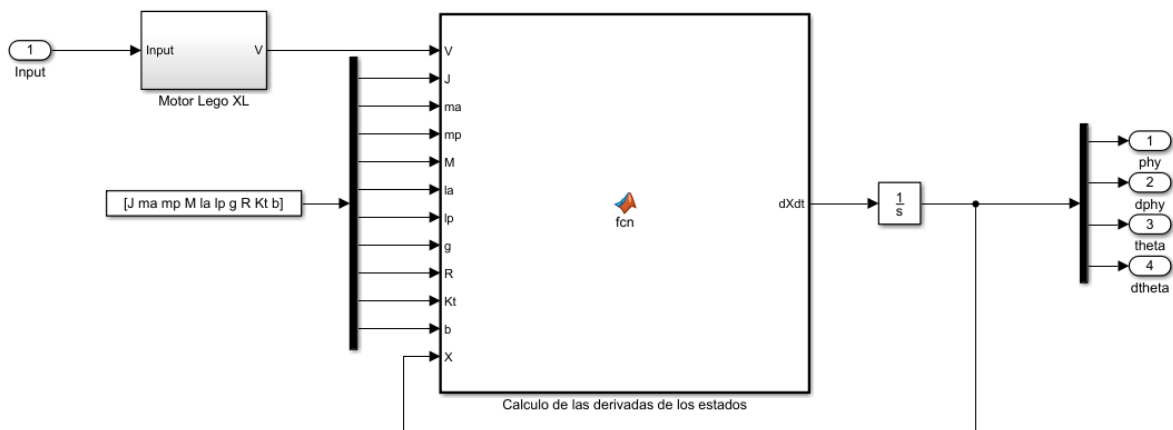


Ilustración 6. Subsistema enmascarado del modelo del péndulo en Simulink

El subsistema que se muestra Ilustración 7 sirve para particularizar el modelo para el motor de Lego *Power Functions XL*. El motor utilizado en la maqueta solo acepta como entrada valores entre 100 y -100. 100 para suministrar la máxima potencia, 0 para no suministrar potencia y -100 para la máxima potencia en sentido inverso. 100 corresponde a 7,4 V de tensión entre los bornes del motor. Se trata de una relación directamente proporcional.

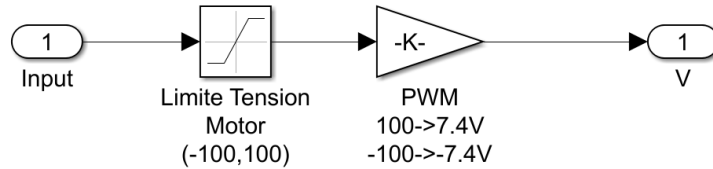


Ilustración 7. Subsistema Motor Lego XL

A continuación se muestra el código que contiene el bloque *MATLAB Function*. En él se realiza el cálculo de las derivadas de las variables de estados. Dentro de este bloque también se calculan los parámetros auxiliares utilizados en el cálculo de las derivadas. En este caso se introduce la ecuación (4.43), que relaciona el par suministrado por el motor con el voltaje aplicado en los bornes del mismo; se ha considerado que la constante de par motor y la de fuerza electromotriz son aproximadamente iguales.

```
function dXdt = fcn(V, J, ma, mp, M, la, lp, g, R, Kt, b, X)

x2=X(2); %-> x1=phy, x2=dphy, x3=theta, x4=dtheta
x3=X(3);
x4=X(4);

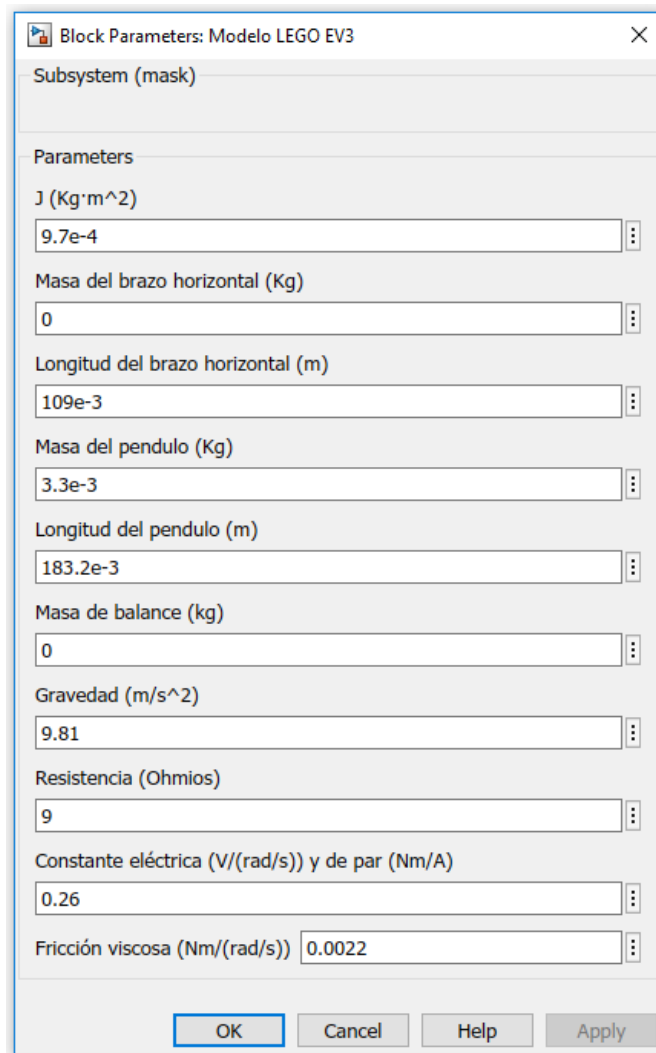
alpha = J+(1/3*ma+mp+M)*la^2; %-> Parametros auxiliares
beta = (1/3*mp+M)*lp^2;
gamma = (1/2*mp+M)*lp*la;
delta = (1/2*mp+M)*lp*g;

tau=(Kt/R)*V-(Kt^2/R)*x2;

dx1=x2;
dx2=(beta*(tau-b*x2)-beta*gamma*cos(x3)^2*sin(x3)*x2^2-2*beta^2*cos(x3)*sin(x3)*x2*x4...
+beta*gamma*sin(x3)*x4^2-gamma*delta*cos(x3)*sin(x3))...
/(alpha*beta-gamma^2+(beta^2+gamma^2)*sin(x3)^2);
dx3=x4;
dx4=(beta*(alpha+beta*sin(x3)^2)*cos(x3)*sin(x3)*x2^2+2*beta*gamma*cos(x3)^2*sin(x3)*x2*x4...
-gamma^2*cos(x3)*sin(x3)*x4^2+delta*(alpha+beta*sin(x3)^2)*sin(x3)-gamma*cos(x3)*(tau-b*x2))...
/(alpha*beta-gamma^2+(beta^2+gamma^2)*sin(x3)^2);

dXdt = [dx1;dx2;dx3;dx4];
```


Crear una máscara permite que el subsistema se convierta en un bloque como cualquier otro de *Simulink* permitiendo así introducir los valores en él como si fuese un bloque cualquiera. También permite hacer una descripción del bloque o que pueda aparecer una imagen o dibujo. En la Ilustración 8 se puede observar la ventana que se obtiene si se pulsa el botón izquierdo sobre el subsistema enmascarado.



Block Parameters: Modelo LEGO EV3

Subsystem (mask)

Parameters

J (Kg·m²)
9.7e-4

Masa del brazo horizontal (Kg)
0

Longitud del brazo horizontal (m)
109e-3

Masa del pendulo (Kg)
3.3e-3

Longitud del pendulo (m)
183.2e-3

Masa de balance (kg)
0

Gravedad (m/s²)
9.81

Resistencia (Ohmios)
9

Constante eléctrica (V/(rad/s)) y de par (Nm/A)
0.26

Fricción viscosa (Nm/(rad/s)) 0.0022

OK Cancel Help Apply

Ilustración 8. Parámetros del modelo del péndulo enmascarado

MEMORIA

Para crear el enmascaramiento del subsistema es necesario, una vez creado el modelo y el subsistema, pulsar sobre el subsistema con el botón derecho sobre el modelo y seleccionar **Mask > Create Mask**. Una vez creada la máscara aparece una ventana como se muestra en la Ilustración 9.

Las dos pestañas que tienen mayor importancia dentro de esta ventana son: **Documentation** que permite dar una descripción del bloque y **Parameters & Dialog** donde se pueden introducir los parámetros que se muestran en el bloque al clicar sobre el cómo en la Ilustración 8. Si es necesario introducir algún parámetro adicional, basta con pulsar **edit**.

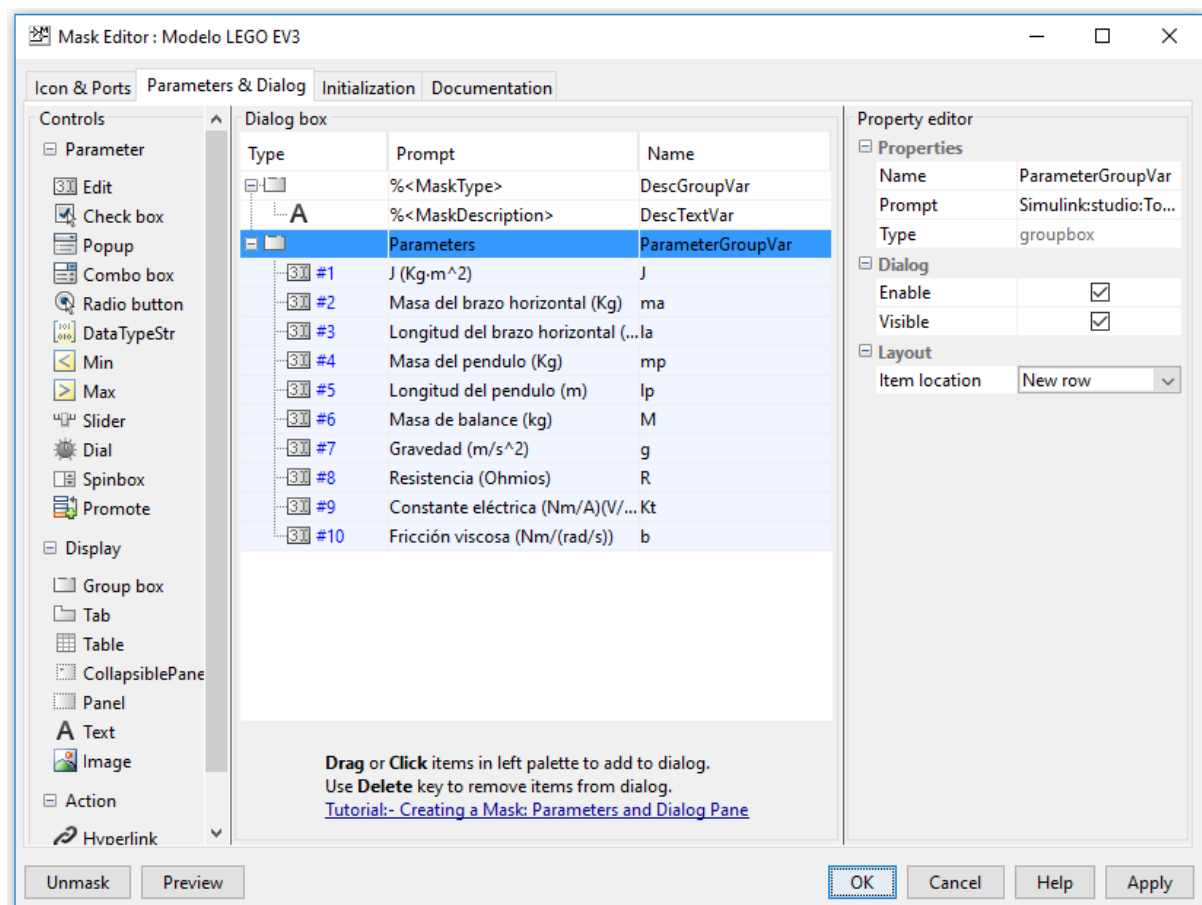


Ilustración 9. Editor de la máscara

MEMORIA

Los parámetros que se muestran en la Tabla 1 serán los valores utilizados en las simulaciones del péndulo que se van a realizar.

PARÁMETRO	VALOR
Masa del brazo horizontal (m_a)	0 kg
Longitud del brazo horizontal (l_a)	$109 \cdot 10^{-3}$ m
Masa del péndulo (m_p)	$3,3 \cdot 10^{-3}$ kg
Longitud del péndulo (l_p)	$183,2 \cdot 10^{-3}$ m
Masa de balance (M)	0 kg
Gravedad (g)	$9,81 \text{ m/s}^2$

Tabla 1. Parámetros del péndulo invertido rotatorio

Los parámetros que se muestran en la Tabla 2 son los valores de los parámetros del motor de corriente continua, cuya obtención se detalla en el Anexo 3.

PARÁMETRO	VALOR
Momento de inercia total (J_m)	$9,7 \cdot 10^{-4} \text{ kg} \cdot \text{m}^2$
Resistencia (R)	9 Ω
Constante de par (K_t)	0,26 N·m/A
Constante de fuerza contraelectromotriz (K_e)	0,26 V·s/rad
Fricción viscosa (b)	0,0022 Nm/(rad/s)

Tabla 2. Parámetros del motor de corriente continua

MEMORIA

Además de los parámetros anteriores y de las condiciones iniciales consideradas en el integrador, para que el péndulo no parta de una posición completamente vertical, si se aplica un voltaje de 0 V al péndulo, se obtiene la respuesta del péndulo en caída libre como se observa en la Ilustración 10.

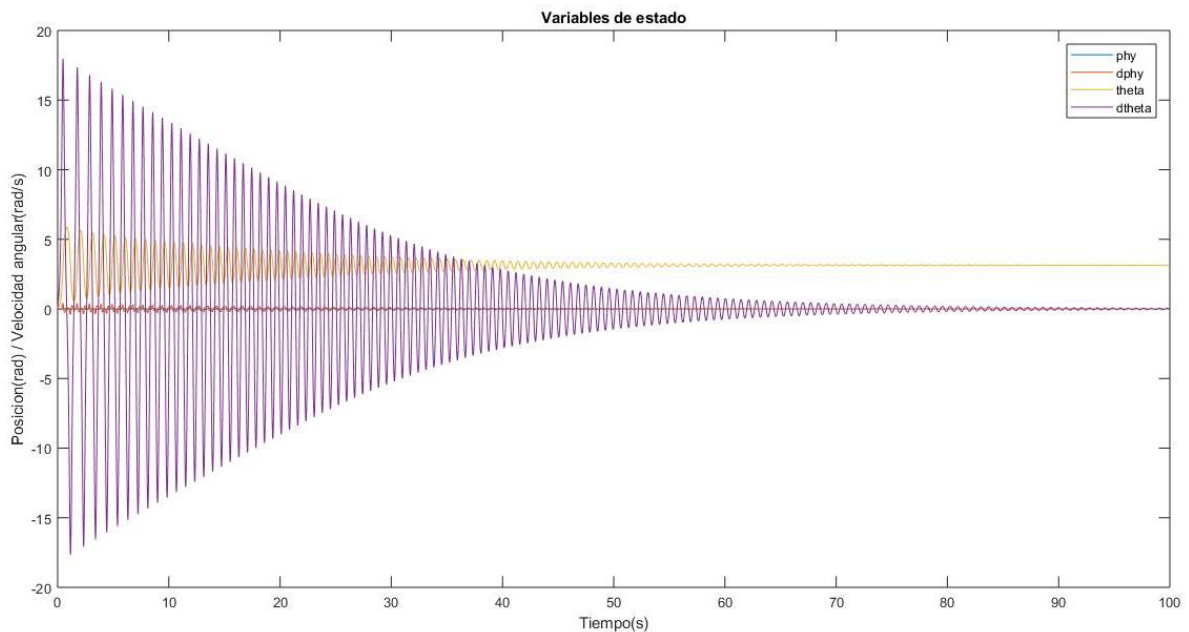


Ilustración 10. Gráficas del movimiento del péndulo

La Ilustración 10 como se puede apreciar, se ha presentado en *Matlab* y no en *Simulink*. Para ello es necesario en el bloque *Scope*, seleccionando **Configuration Properties > Logging > Log data to workspace** con **Variable name:** *datos_pendulo* y **Save format:** *Structure With Time*, para poder almacenar los datos de la simulación en el **Workspace**. Como se ve en la Ilustración 11.

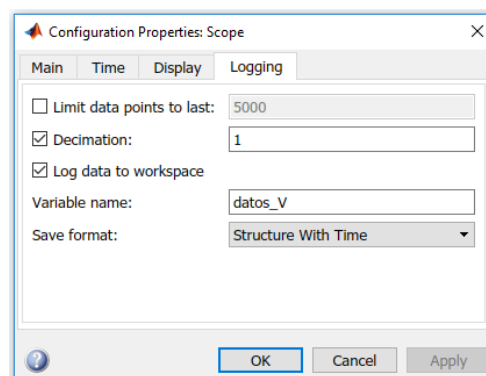


Ilustración 11. Configuración del bloque Scope

Con un script en *Matlab* como el siguiente se visualiza la Ilustración 10.

```
phy = datos_pendolo.signals(1).values;  
dphy = datos_pendolo.signals(2).values;  
theta = datos_pendolo.signals(3).values;  
dtheta = datos_pendolo.signals(4).values;  
tiempo = datos_pendolo.time;  
  
figure, plot(tiempo,phy,tiempo,dphy,tiempo,dtheta,tiempo,theta)  
title('Variables de estado')  
xlabel('Tiempo (s)')  
ylabel('Posicion (rad) / Velocidad Angular (rad/s)')
```

Como se puede observar, visualizar el movimiento del péndulo a partir de la Ilustración 10 puede resultar complicado. Tanto la posición (*phy*) como la velocidad angular del motor (*dphy*) es prácticamente cero, mientras que la posición del péndulo (*theta*) se inicia en 0.1 radianes para terminar estabilizándose en π radianes que representa la posición vertical inferior del péndulo. La velocidad angular del péndulo (*dtheta*) se va reduciendo progresivamente hasta 0 rad/s, una vez que el péndulo se ha estabilizado en la posición vertical inferior.

6. Diseño del controlador

Introducción

Un sistema físico es complejo, debido a que posee gran cantidad de entradas y salidas que están interrelacionadas generalmente de forma no lineal, como es el caso de péndulo de Furuta.

El péndulo de Furuta es un sistema sub-actuado, que además también posee no linealidades, razón por la que se debe garantizar mediante una ley de control la estabilidad del sistema en lazo cerrado en el punto de operación.

La teoría básica de sistemas de control normalmente está enfocada al análisis de la relación entrada-salida o función de transferencia. La realimentación de las variables de estado puede ser usada para modificar las frecuencias naturales del sistema, y en particular, hacerlas todas estables, siempre que el sistema sea controlable por realimentación de variables de estado.

Mediante la realimentación de las variables de estado es posible dar solución al problema de regulación del sistema. El problema de regulación consiste en llevar todas las variables de estado del sistema desde un estado diferente de cero, debido a condiciones iniciales o a la influencia de perturbaciones, hasta el valor de cero, que representa el punto de equilibrio deseado.

Regulación mediante realimentación de variables de estado

Para el diseño de un controlador por realimentación de variables de estado, es necesario, obtener el modelo linealizado de la planta alrededor de un punto de equilibrio. Este modelo se puede observar en (4.69).

Se considera el sistema descrito mediante variables de estado, como se muestra en las ecuaciones (6.1) y (6.2)

$$\dot{x}(t) = Ax(t) + Bu(t) \quad (6.1)$$

$$y(t) = Cx(t) + Du(t) \quad (6.2)$$

Con polinomio característico definido por la ecuación (6.3).

$$a(s) = \det(sI - A) = s^n + a_1s^{n-1} + \dots + a_{n-1}s + a_n \quad (6.3)$$

MEMORIA

Es de interés modificar las frecuencias naturales del sistema, es decir, los valores propios mediante la realimentación de las variables de estado para obtener un sistema estable con unos parámetros de respuesta en el tiempo acorde con los criterios de diseño. Para lograr este objetivo se define la ley de control (6.4).

$$u(t) = -Kx(t) \quad (6.4)$$

Sustituyendo la ecuación (6.4) en la expresión (6.1), se obtiene (6.5).

$$\dot{x}(t) = Ax(t) - BKx(t)$$

$$\dot{x}(t) = (A - BK)x(t) \quad (6.5)$$

En la Ilustración 12 se muestra el esquema de realimentación para regulación y donde se puede notar que este esquema no considera un comando de referencia, ya que en el sistema del péndulo rotario invertido la referencia siempre será cero, el objetivo es estabilizar el péndulo en el punto de equilibrio, a pesar de perturbaciones que lo desvíen o condiciones iniciales distintas de cero.

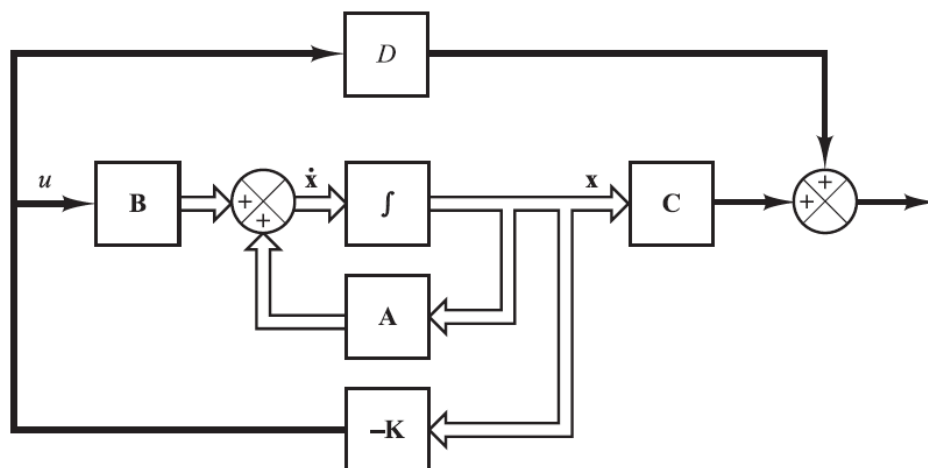


Ilustración 12. Sistema de control en lazo cerrado para regulación

La dinámica resultante tiene como polinomio característico la expresión (6.6), donde se puede observar que las ganancias de realimentación **K** intervienen directamente en la ubicación de los polos de lazo cerrado.

$$a(s) = \det(sI - A + BK) \quad (6.6)$$

Controlabilidad y observabilidad

Esta sección aborda la *controlabilidad* y la *observabilidad* de un sistema en el espacio de estados.

Se dice que un sistema es *controlable* en el tiempo t_0 si se puede transferir desde cualquier estado inicial $x(t_0)$ a cualquier otro estado, mediante un vector de control sin restricciones, en un intervalo de tiempo finito.

Se dice que un sistema es *observable* en el tiempo t_0 si, con el sistema en el estado $x(t_0)$, es posible determinar este estado a partir de la observación de la salida durante un intervalo de tiempo finito.

Los conceptos de *controlabilidad* y *observabilidad* juegan un papel importante en el diseño de un sistema de control en el espacio de estados. De hecho las condiciones de controlabilidad y observabilidad determinan la existencia de una solución completa para un problema de diseño de un sistema de control. La solución a este problema puede no existir si el sistema considerado no es controlable. Aunque la mayor parte de los sistemas físicos son controlables y observables, los modelos matemáticos correspondientes tal vez no posean la propiedad de controlabilidad y observabilidad. En este caso, es necesario conocer las condiciones en las cuales un sistema es controlable y observable.

Controlabilidad

Sea el sistema en tiempo continuo descrito en la ecuación (6.1), donde:

x = vector de estados (vector de dimensión n)

u = vector de entradas (vector de dimensión r)

A = matriz de $n \times n$

B = matriz de $n \times r$

Se dice que el sistema descrito por la ecuación (6.1) es de estado controlable en $t=t_0$, si es posible construir una señal de control sin restricciones que transfiera un estado inicial a cualquier estado final en intervalo de tiempo finito $t_0 \leq t \leq t_1$. Si todos los estados son controlables se dice que el sistema es de estado completamente controlable.

MEMORIA

Si el sistema se describe por la ecuación (6.1) donde u es un vector de dimensión r , se demuestra que la condición para la controlabilidad completa del estado es que la matriz $n \times nr$, que se observa en (6.7), sea de rango n , o que contenga n vectores columna linealmente independientes. La matriz (6.7) se conoce comúnmente como *matriz de controlabilidad*.

$$M_c = [B : AB : A^2B : \dots : A^{n-1}B] \quad (6.7)$$

Observabilidad

Sea el sistema no forzado descrito en las ecuaciones (6.8) y (6.9).

$$\dot{x}(t) = Ax(t) \quad (6.8)$$

$$y(t) = Cx(t) \quad (6.9)$$

Donde:

x = vector de estados (vector de dimensión n)

y = vector de salida (vector de dimensión m)

A = matriz de $n \times n$

C = matriz de $m \times n$

Se dice que el sistema es completamente observable si el estado $x(t_0)$ se determina a partir de la observación de $y(t)$ durante un intervalo de tiempo finito, $t_0 \leq t \leq t_1$. Por tanto, el sistema es completamente observable si todas las transiciones del estado afectan eventualmente a todos los elementos del vector de salida. El concepto de observabilidad es útil al resolver el problema de reconstruir variables de estado no medibles a partir de variables que sí lo son en el tiempo mínimo posible.

El concepto de observabilidad es muy importante porque, en la práctica, la dificultad que se encuentra con el control mediante realimentación del estado es que algunas de las variables de estado no son accesibles para una medición directa, por lo que se hace necesario estimar las variables de estado no medibles para construir las señales de control.

MEMORIA

Se puede expresar la condición para observabilidad completa del siguiente modo. El sistema descrito por las ecuaciones (6.8) y (6.9) es completamente observable si y solo si la matriz $n \times nm$, que se observa en (6.10), es de rango n , o tiene n vectores columna linealmente independientes. Esta matriz (6.10) se denomina *matriz de observabilidad*.

$$M_o = [C^T : A^T C^T : (A^T)^2 C^T : \dots : (A^T)^{n-1} C^T] \quad (6.10)$$

Cálculo de las ganancias de realimentación

Existen diversas metodologías para la obtención de los parámetros de realimentación K en (6.4), y su aplicación depende de los criterios de diseño y requerimientos del sistema, aunque la complejidad puede aumentar con la utilización de uno u otro método, por lo que depende de la consideración del diseñador la selección del mismo. Dos de las dos metodologías más habituales son: *asignación de polos* y *regulador lineal óptimo cuadrático (LQR)*. En este caso se va a utilizar el segundo método.

El control *LQR* permite para un sistema definido en el espacio de estados encontrar las ganancias de realimentación óptimas considerando algunos criterios de optimización, como son la ubicación de los polos y las magnitudes máximas permisibles tanto en la señal de control, como en las variables de estado.

Se dice que es una estrategia de control óptima ya que se procura la operación del sistema dinámico al menor coste, es decir que los ajustes del controlador proporcional se encuentran usando un algoritmo matemático que minimiza la función de coste o desviaciones no deseadas.

Formulación del LQR

Considerando el modelo de espacio de estados definido en la expresión (4.69), en el cual se supone que la matriz de controlabilidad M_c tiene rango n , se plantea la integral cuadrática (6.11).

$$J = \frac{1}{2} x^T(t_f) P_x^T(t_f) + \frac{1}{2} \int_{t_0}^{t_f} (x^T Q x + u^T R u) dt \quad (6.11)$$

MEMORIA

La expresión (6.11) se puede simplificar, si se considera $t_f \rightarrow \infty$, el estado $x(t_f) \rightarrow 0$ para sistemas de regulación, siendo reescrita como se muestra en (6.12), en donde J es conocida como la función de coste.

$$J = \frac{1}{2} \int_{t_0}^{\infty} (x^T Q x + u^T R u) dt \quad (6.12)$$

El control LQR minimiza la función de coste J según la ecuación (6.12), en la que:

- Q y R son matrices $n \times n$, $m \times m$ positivas definidas, donde n es el número de variables de estado y m es el número de entradas.
- La cantidad $x^T Q x$ es una medida de la desviación del estado, con respecto al estado cero, en un tiempo t y $u^T R u$ es una medida del esfuerzo de la señal de control, en un tiempo t ; la integral obtiene como resultado el acumulado de las medidas anteriores.
- Las matrices Q y R , son determinadas por el diseñador de forma empírica y cuyos elementos deben ser positivos o cero. Estos valores expresan la penalización en J de las desviaciones de los estados y de la acción de control empleada. En general el diseño de un controlador LQR óptimo, depende de la habilidad en la minimización de la función de coste, lo cual supone que para obtener un mayor grado de fidelidad con respecto a los requerimientos del sistema, es necesario un mayor conocimiento de la planta.

Obtención de las ganancias de realimentación y ecuación de Riccati

Definiendo la ley de control óptima como (6.13) para el sistema en espacio de estados definido por (6.1) y teniendo en cuenta que la ganancia de realimentación $K(t)$ para un controlador LQR tiene la forma (6.14), es posible encontrar el valor de dicha ganancia mediante el previo conocimiento de la matriz $P(t)$, cuya solución es hallada desarrollando la ecuación matricial de *Riccati* (6.15), considerando condiciones iniciales.

$$u(t) = -K(t)x(t) \quad (6.13)$$

$$K(t) = R^{-1} B^T P(t) \quad (6.14)$$

$$\dot{P} = A^T P + P A + Q - P B R^{-1} B^T P \quad (6.15)$$

MEMORIA

Si para la ecuación (6.15), las matrices **A**, **B**, **Q** y **R**, son independientes del tiempo y $t_f \rightarrow \infty$, entonces $\dot{P} = 0$, con lo que la ecuación matricial de *Riccati* se convierte en la ecuación algebraica de *Riccati*, como se puede ver en (6.16).

$$A^T P + PA + Q - PBR^{-1}B^T P = 0 \quad (6.16)$$

La ecuación de *Riccati* puede resolverse solo si el sistema es controlable y observable, por lo tanto antes de aplicar la ley de control LQR, debe comprobarse que el proceso a controlar cumpla con estas dos condiciones. Resolviendo la ecuación algebraica de *Riccati*, es posible encontrar la ganancia de realimentación **K**, que será más o menos óptima en la medida que las matrices **Q** y **R**, hayan sido bien escogidas por el diseñador.

Como se puede ver en las ecuaciones (6.13), (6.14) y (6.16) contienen los términos **Q** y **R** estas son matrices propias del control LQR. Hay algunos parámetros que deben tomarse en cuenta para seleccionar estas matrices, sin embargo no existe una regla determinada para hacerlo, además la estrategia LQR tiene la ventaja de que independientemente de la selección de **Q** y **R** sigue siendo capaz de estabilizar al sistema porque es un control robusto. Algunos parámetros que suelen seguirse para seleccionar estas matrices son:

- Los valores deben ser siempre positivos o cero.
- Debe conocerse muy bien el comportamiento dinámico de la planta.
- Con frecuencia se escoge **Q** y **R** diagonales asignando valores grandes a las variables que se quieran minimizar.
- Si **Q** contiene valores grandes la exigencia de control será mayor.
- **Q** debe ser positiva semidefinida, es decir que cumpla con $Q = Q^T \geq 0$.
- **R** debe ser positiva definida, es decir que cumpla con $R = R^T > 0$.
- Si **R** contiene valores grandes, se incrementa la influencia de las acciones de control sobre la planta.
- La manera más fácil de escoger **Q** es: $Q = C^T C$.

Implementación LQR

Para utilizar esta estrategia de control es necesario obtener un modelo lineal en el espacio de estados del péndulo invertido rotatorio. Este modelo está definido en (4.69).

Utilizando los parámetros de la Tabla 1 y de la Tabla 2; además de utilizar un script de *Matlab* para definir el modelo lineal del péndulo es posible obtener los valores de las matrices **A**, **B**, **C** y **D**.

```
% Parametros
J = 9.7e-4;           %->Momento de inercia (kg·m^2)
ma = 0;              %->Masa del brazo horizontal (kg)
la = 109e-3;          %->Longitud del brazo horizontal (m)
mp = 3.3e-3;          %->Masa del pendulo (kg)
lp = 183.2e-3;        %->Longitud del pendulo (m)
M = 0;               %->Masa de balance (kg)
g = 9.81;             %->Gravedad (m/s^2)
r = 9;               %->Resistencia del motor (ohmios)
Kt = 0.26;           %->Constante electrica del motor (V/(rad/s))
b=0.0022;            %->Friccion viscosa

% Parametros auxiliares
alpha = J+(1/3*ma+mp*M)*la^2;
beta = (1/3*mp*M)*lp^2;
gamma = (1/2*mp*M)*lp*la;
delta = (1/2*mp*M)*lp*g;

% Matrices -> Punto de equilibrio X0=(0,0,0,0) PWM (-100,100)
A = [0 1 0 0; 0 -(beta*(Kt^2+r*b))/(r*(beta*alpha-gamma^2)) -(delta*gamma)/(beta*alpha-gamma^2) 0; ...
     0 0 0 1; 0 (gamma*(Kt^2+r*b))/(r*(beta*alpha-gamma^2)) (alpha*delta)/(beta*alpha-gamma^2) 0];
% (4x4)
B = [0; Kt*beta/(r*(beta*alpha-gamma^2)); 0; -Kt*gamma/(r*(beta*alpha-gamma^2))]*7.4/100; % (4x1)
C = [1 0 0 0; 0 0 1 0]; % (2x4)
D = [0; 0]; % (2x1)
% Espacio de estados -> dx=A*x+B*u (4x1) / y=C*x*D*u (2x1)
Pss = ss(A,B,C,D); %Espacio de estados
Ptf = zpk(tf(Pss)); %Funcion de transferencia
pole(Pss);
```

Evaluando los parámetros de la Tabla 1 y de la Tabla 2. Parámetros del motor de corriente continua; y los parámetros auxiliares definidos en (4.31) en las matrices **A** y **B** mostradas en (4.69), se obtiene (6.17):

$$A = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & -9,9113 & -2,701 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 8,8455 & 82,7327 & 0 \end{pmatrix} \quad (6.17)$$

$$B = \begin{pmatrix} 0 \\ 2,1818 \\ 0 \\ -1,9472 \end{pmatrix}$$

Una vez se obtienen los valores reales de las matrices que forman el modelo de estado del sistema, se genera un pequeño script también en *Matlab* para comprobar la controlabilidad y observabilidad del sistema.

```
%% Controlabilidad y observabilidad
Co = ctrb(Pss);           %->Matriz de controlabilidad
unco = length(A)-rank(Co);
display(unco)
Ob = obsv(Pss);          %->Matriz de observabilidad
unob = length(A)-rank(Ob);
display(unob)
```

La instrucción *ctrb* de *Matlab* calcula la matriz de controlabilidad, restando a la dimensión de la matriz **A**, el rango de la matriz de controlabilidad, se obtiene la diferencia de ambas. Si la diferencia es cero el sistema es controlable.

La instrucción *obsv* de *Matlab* calcula la matriz de observabilidad, restando a la dimensión de la matriz **A**, el rango de la matriz de observabilidad, se obtiene la diferencia de ambas. Si la diferencia es cero el sistema es observable.

Cumpliendo ambas condiciones se puede proceder al diseño del controlador LQR. Para ello se utiliza la instrucción *lqr* de *Matlab*, esta instrucción tiene la siguiente forma:

$[K, S, e] = \text{lqr}(A, B, Q, R, N);$

Las matrices **A** y **B** son las dos matrices que definen el sistema; las matrices **Q** y **R** son de ponderación, aparecen en la función de coste definida en (6.12) y **N** es un término que se va omitir. Las salidas que genera la función son **K** (la ganancia óptima), **S** (la solución de la ecuación asociada de *Riccati*) y **e** (los polos del sistema en lazo cerrado). Con un script como el siguiente se obtiene la ganancia óptima.

```
%% Control LQR
Q = [q1 0 0 0; 0 q2 0 0; 0 0 q3 0; 0 0 0 q4];
R = rl;
[K,S,e] = lqr(A,B,Q,R);
fprintf('La ganancia optima es:')
display(K)
fprintf('Los polos:')
display(e)
```

En el siguiente apartado se utiliza el modelo creado en *Simulink* para probar los controladores y realizar la optimización para buscar el mejor controlador, variando las matrices de ponderación **Q** y **R** en el anterior script.

Optimización de la ley de control

Utilizando el modelo implementado en *Simulink*, se puede diseñar en la misma plataforma un esquema de control para probar y optimizar los controladores. En la Ilustración 13 se puede observar tanto el modelo no lineal, como el lazo de control creado para desarrollar el modelo del péndulo invertido rotatorio en el punto de equilibrio como en la Ilustración 12.

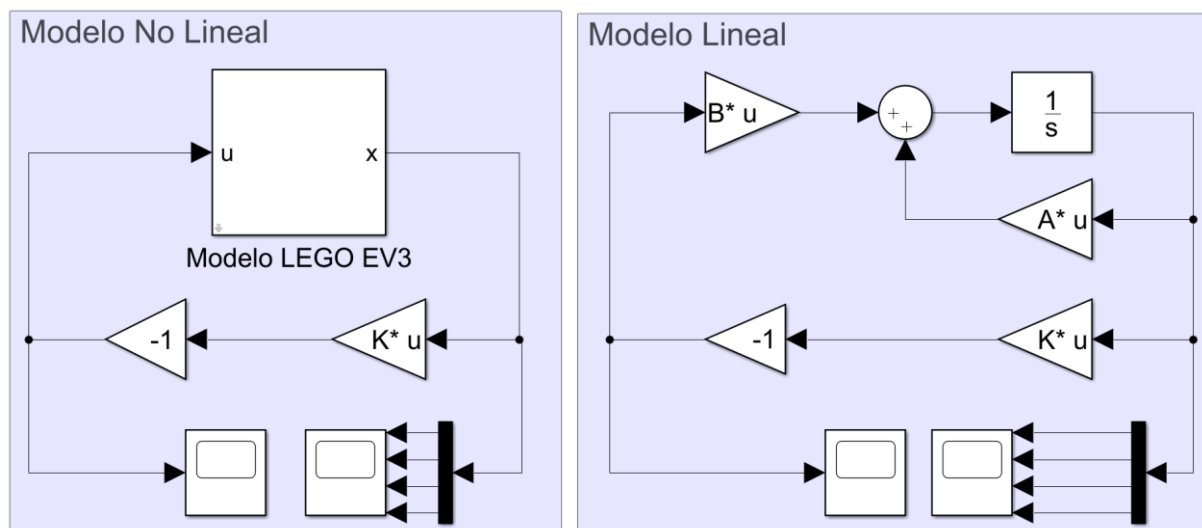


Ilustración 13. Implementación del controlador LQR en Simulink

Los bloques de ganancias que se han introducido no son las habituales, son de tipo *Matrix* ($K*u$). Las ganancias K representan la obtenida mediante el control LQR y las ganancias A y B provienen del modelo de estado linealizado en torno al punto de equilibrio seleccionado del sistema. La ganancia inversora, junto con la K , implementa la ley de control (6.4). El integrador obtiene las variables de estado y además establece las condiciones iniciales del modelo lineal.

Se va a comprobar el comportamiento de los controladores frente a las condiciones iniciales que sitúan el péndulo fuera del punto de equilibrio. El ángulo del motor, la velocidad angular del motor y la velocidad angular del péndulo serán cero, mientras que el ángulo del péndulo será de 0,1 radianes.

Utilizando el script anterior se van a diseñar seis controladores diferentes, modificando las matrices de ponderación. Todas las gráficas de simulación que se van a mostrar a continuación corresponden al modelo no lineal, ya que ambos modelos (lineal y no lineal) se comportan de la misma forma cuando funcionan en torno al punto de equilibrio. Aunque se presentan gráficas de todas las variables de estado del sistema, se presta especial importancia a las gráficas de los ángulos y de la acción de control, que representan las salidas y la entrada al sistema respectivamente.

MEMORIA

En cada controlador se observan las matrices de ponderación utilizadas (Q y R), la ganancia obtenida (K) y los polos en lazo cerrado del sistema (e). El controlador inicial se diseña de la forma $Q = C^T C$. En los siguientes controladores, la matriz R se mantiene constante, mientras que en la matriz Q se aumentan los pesos, que penalizan las desviaciones de los ángulos en la función de coste J , lo que implicará leyes de control u que traten de devolver estos estados más rápidamente al equilibrio.

Controlador 1

$$Q = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \quad (6.18)$$

$$R = 1$$

$$K = (-1 \quad -9,4096 \quad -179,8274 \quad -19,9260)$$

$$e = (-0,02202 \quad -10,6607 \quad -8,2356 \quad -9,0650)$$

Controlador 2

$$Q = \begin{pmatrix} 10 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 10 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \quad (6.19)$$

$$R = 1$$

$$K = (-3,1623 \quad -10,1038 \quad -188,9287 \quad -20,9389)$$

$$e = (-10,4970 \quad -9,3828 \quad -8,0614 \quad -0,6980)$$

Controlador 3

$$Q = \begin{pmatrix} 20 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 20 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \quad (6.20)$$

$$R = 1$$

$$K = (-4,4721 \quad -10,5201 \quad -184,3730 \quad -21,5436)$$

$$e = (-10,1328 \quad -9,8620 \quad -7,9238 \quad -0,9898)$$

Controlador 4

$$Q = \begin{pmatrix} 30 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 30 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \quad (6.21)$$

$$R = 1$$

$$K = (-5,4772 \quad -10,8376 \quad -198,5189 \quad -22,0035)$$

$$e = (-10,0416 \pm 0,4883i \quad -7,8123 \quad -1,2157)$$

Controlador 5

$$Q = \begin{pmatrix} 40 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 40 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \quad (6.22)$$

$$R = 1$$

$$K = (-6,3246 \quad -11,1041 \quad -201,9942 \quad -22,3886)$$

$$e = (-10,0780 \pm 0,6883i \quad -7,7158 \quad -1,4078)$$

Controlador 6

$$Q = \begin{pmatrix} 50 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 50 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \quad (6.23)$$

$$R = 1$$

$$K = (-7,0711 \quad -11,3381 \quad -205,0417 \quad -22,7261)$$

$$e = (-10,1092 \pm 0,8332i \quad -7,6292 \quad -1,5787)$$

MEMORIA

En la Ilustración 14 se puede ver la respuesta del ángulo del motor con los diferentes controladores diseñados. Se observa que a medida que se aumentan los pesos de la matriz Q el sistema responde de forma mas rápida. El Controlador 6 (azul) es mucho más rápido que el controlador inicial (amarillo).

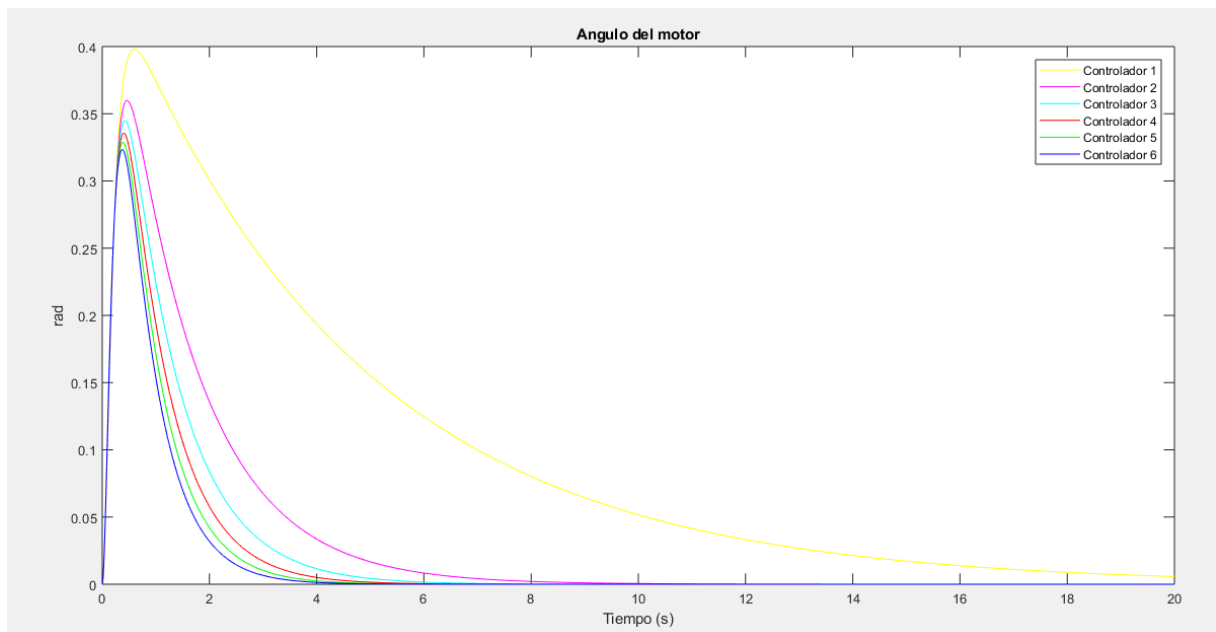


Ilustración 14. Gráfica de la respuesta del ángulo del motor

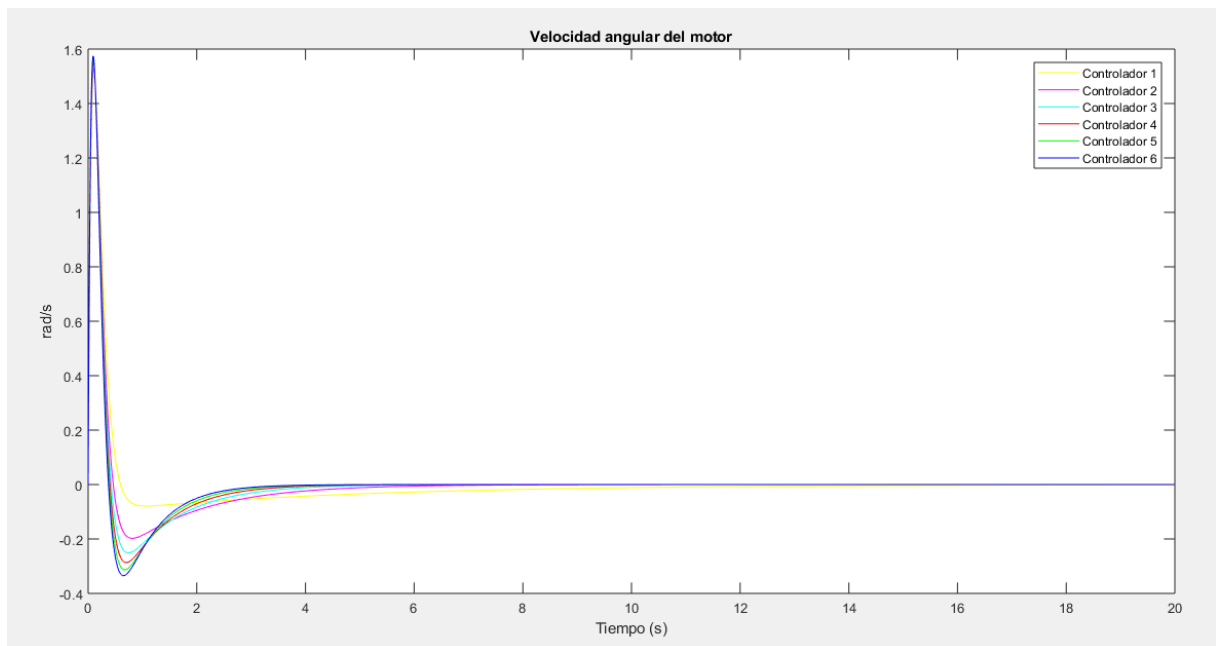


Ilustración 15. Gráfica de la respuesta de la velocidad angular del motor

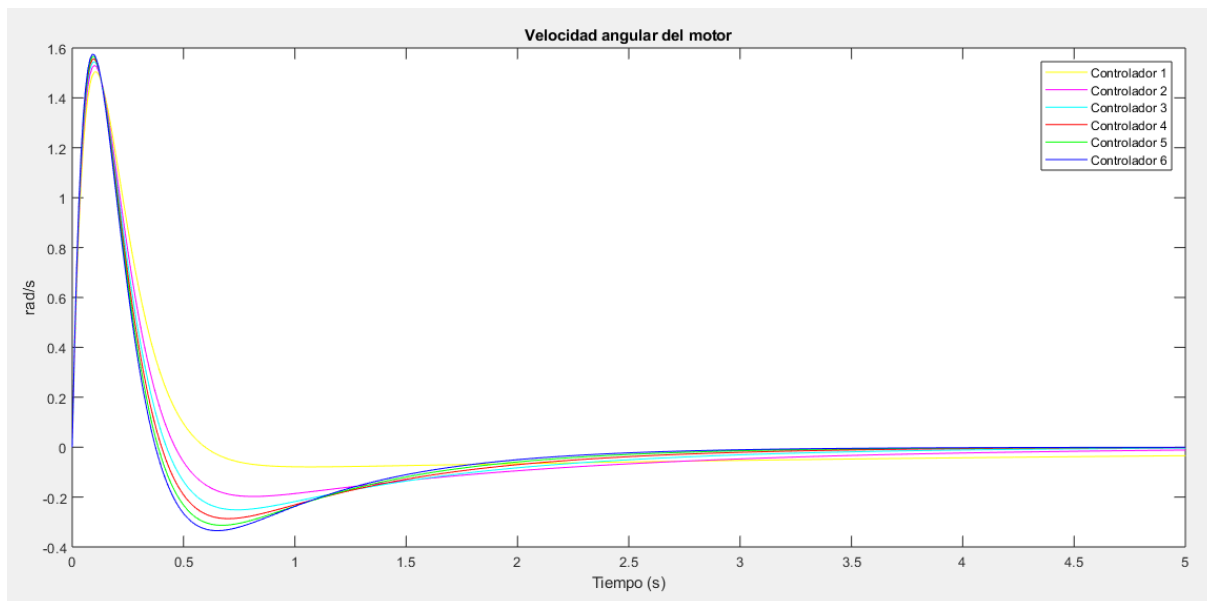


Ilustración 16. Gráfica de la respuesta de la velocidad angular del motor (5 segundos)

En la Ilustración 17 se puede observar la respuesta del ángulo del péndulo con los diferentes controladores diseñados.

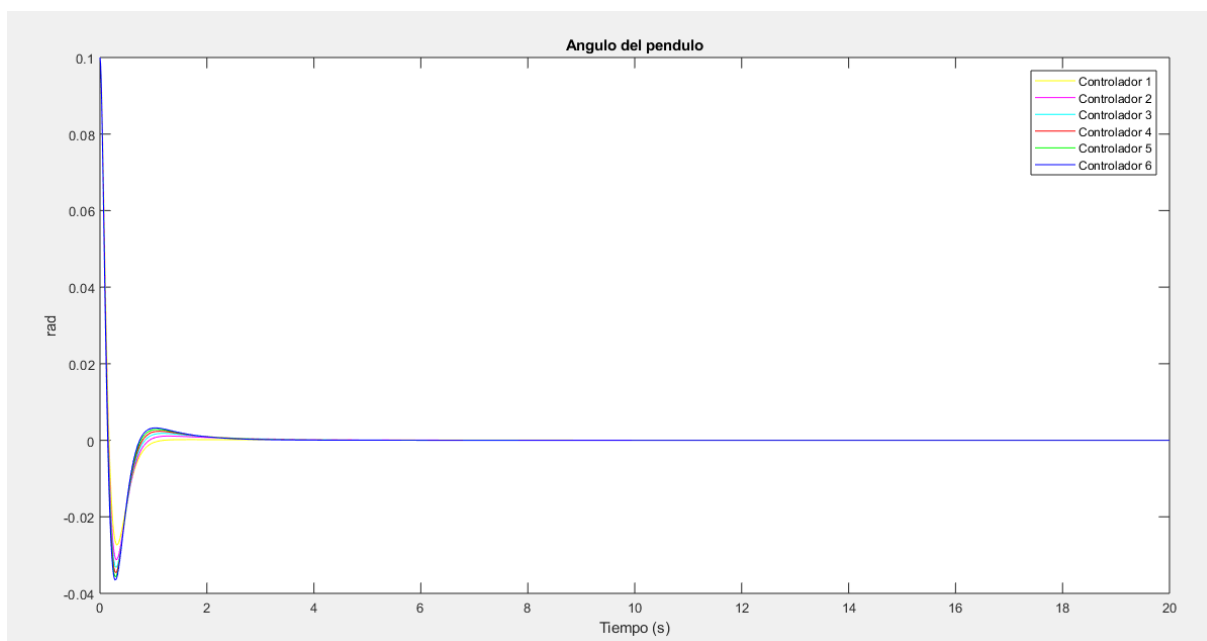


Ilustración 17. Gráfica de la respuesta del ángulo del péndulo

MEMORIA

En la Ilustración 18 se puede ver la respuesta del ángulo del péndulo ampliada a los dos primeros segundos. Se observa que el Controlador 1 (amarillo) es más lento que cualquiera de los siguientes. El Controlador 6 (azul) es el más rápido, pero tiene el mayor sobreimpulso. Aun así cualquiera de las respuestas de los seis controladores es buena.

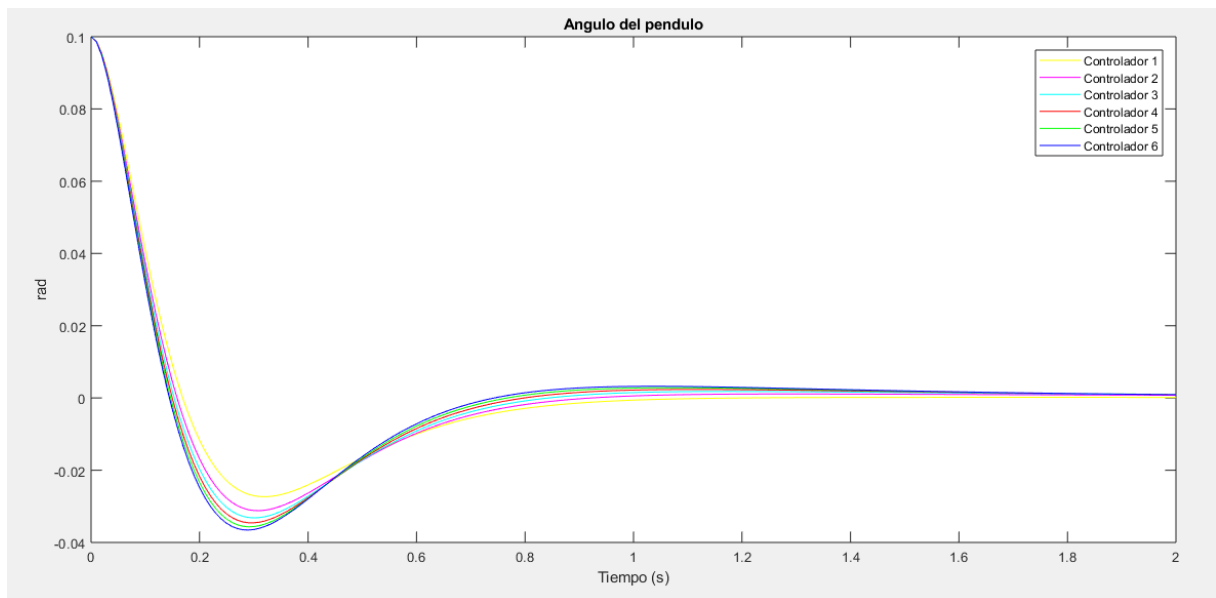


Ilustración 18. Gráfica de la respuesta del ángulo del péndulo (2 segundos)

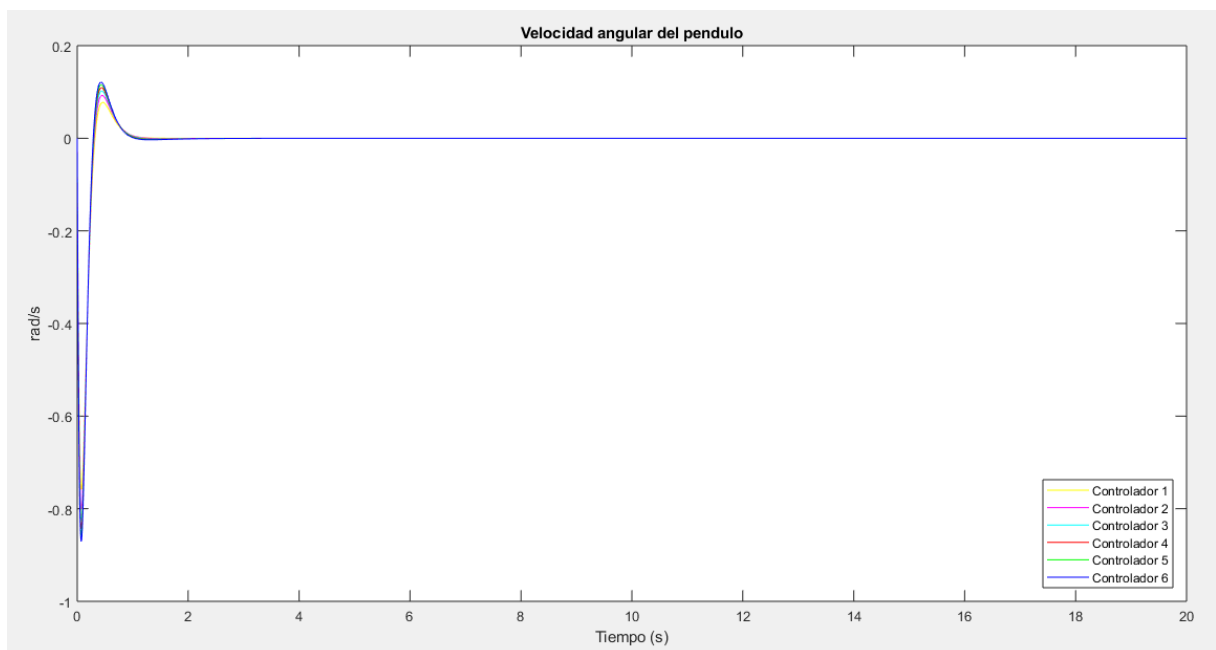


Ilustración 19. Gráfica de la respuesta de la velocidad angular del péndulo

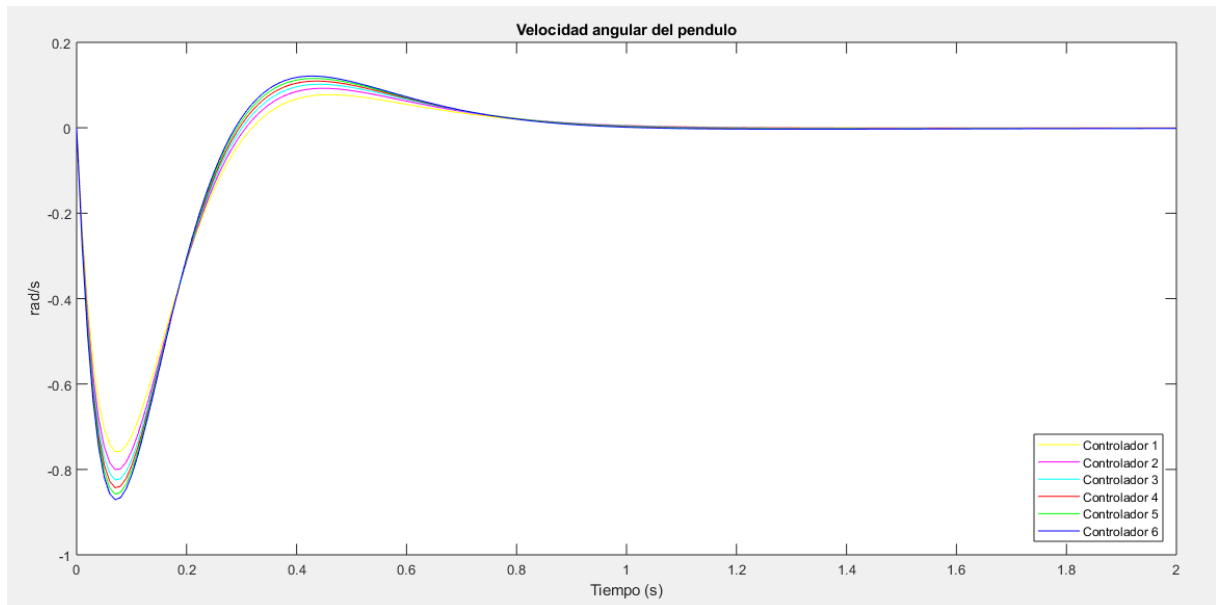


Ilustración 20. Gráfica de la respuesta de la velocidad angular (2 segundos)

Por último, en la Ilustración 21 se puede observar la acción de control suministrada por los diferentes controladores diseñados.

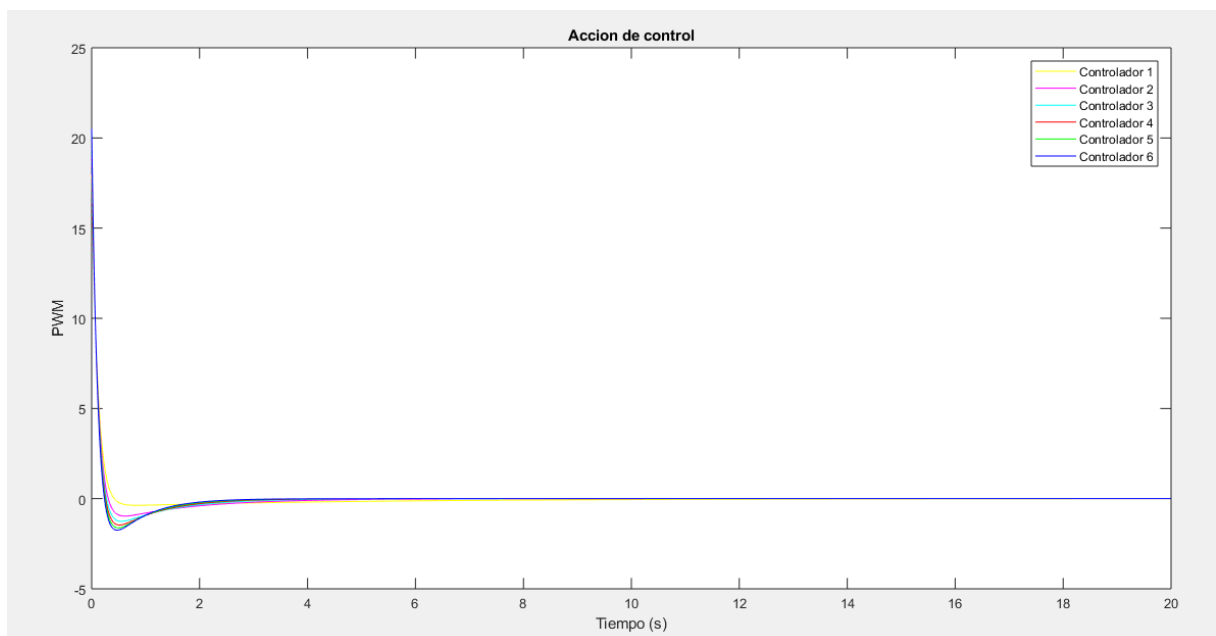


Ilustración 21. Gráfica de la acción de control

MEMORIA

En la Ilustración 22 y en la Ilustración 23 se puede ver la acción de control ampliada a un rango de segundos menor que en la Ilustración 21. Se observa que a medida que aumentan los pesos de la matriz Q la exigencia en la acción de control es mayor, aunque los valores son cercanos entre los 6 controladores.

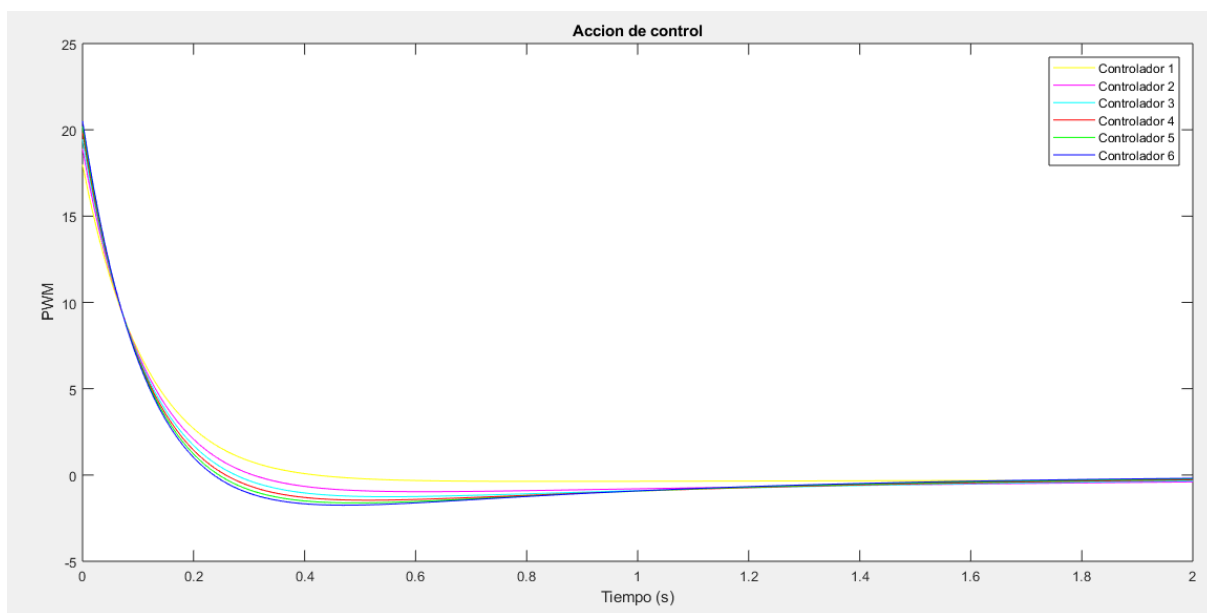


Ilustración 22. Gráfica de la acción de control (2 segundos)

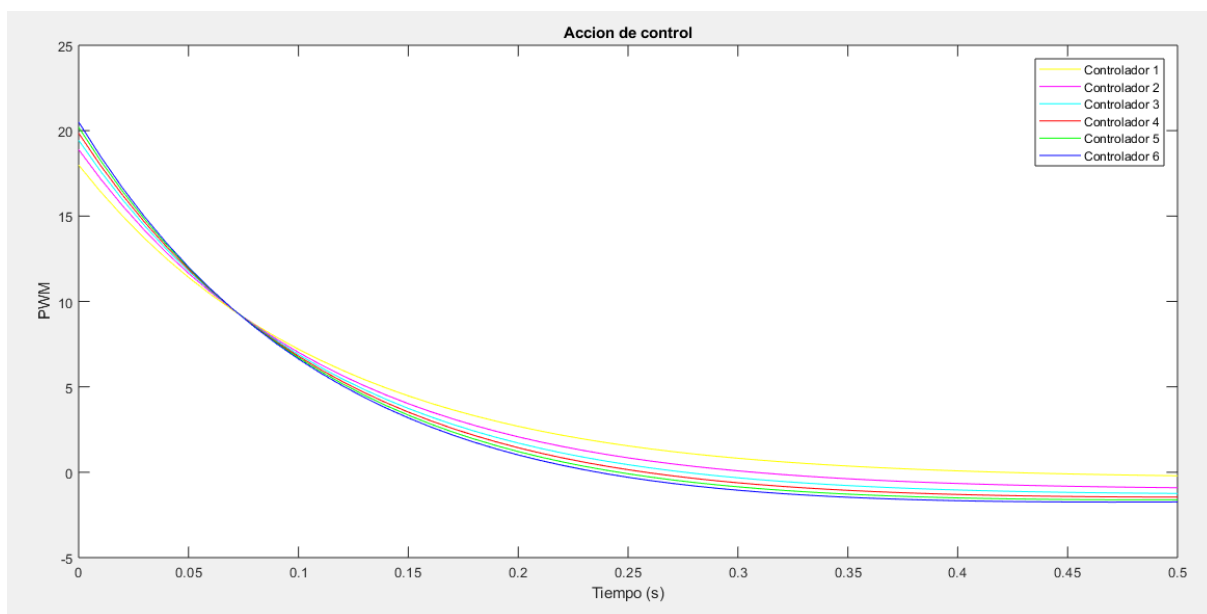


Ilustración 23. Gráfica de la acción de control (0,5 segundos)

7. Control del péndulo rotatorio invertido real

En esta sección se van a diseñar dos modelos de *Simulink* que permitan la comunicación entre el Lego Mindstorms EV3 y el ordenador; y el control del péndulo invertido rotatorio. El modelo de *Simulink* del ordenador intercambiará información con el ladrillo EV3 para monitorizar el movimiento del péndulo y el modelo del péndulo realizará el control del mismo.

En el Anexo 2 se detalla cómo realizar la conexión con el LEGO EV3 a través del ordenador. Además se explicará cómo configurar un modelo de *Simulink* para utilizarlo en dicha plataforma. Por último, se profundiza en cada uno de los bloques utilizados del LEGO EV3 para *Simulink* con más detalle.

Por un lado, se creará un modelo de *Simulink* en el ordenador compuesto por *Dashboard* widgets y bloques de comunicación. Este modelo se utilizará a modo de interface para monitorizar el péndulo (arrancar o parar el actuador, resetear los sensores y modificar las ganancias de realimentación). Además, este interface recibirá desde el LEGO EV3: los datos de movimiento adquiridos por los sensores y la señal de control suministrada al actuador.

Por otro lado el ladrillo EV3 ejecutará un modelo de *Simulink* que recibirá y utilizará los datos enviados desde el ordenador para realizar el control del péndulo invertido rotatorio real. El control del péndulo se realizará en tiempo real desde este modelo. Este modelo será el encargado de: leer los sensores, calcular la ley de control y mandar esta señal de control al motor. Este modelo enviará al modelo del ordenador los datos obtenidos por los sensores, además de la señal de control y el tiempo real transcurrido en el péndulo.

MEMORIA

En la Ilustración 24 se puede observar el modelo de *Simulink* que se va a ejecutar en el LEGO EV3. A continuación se van a explicar todos los subsistemas que forman parte de este modelo.

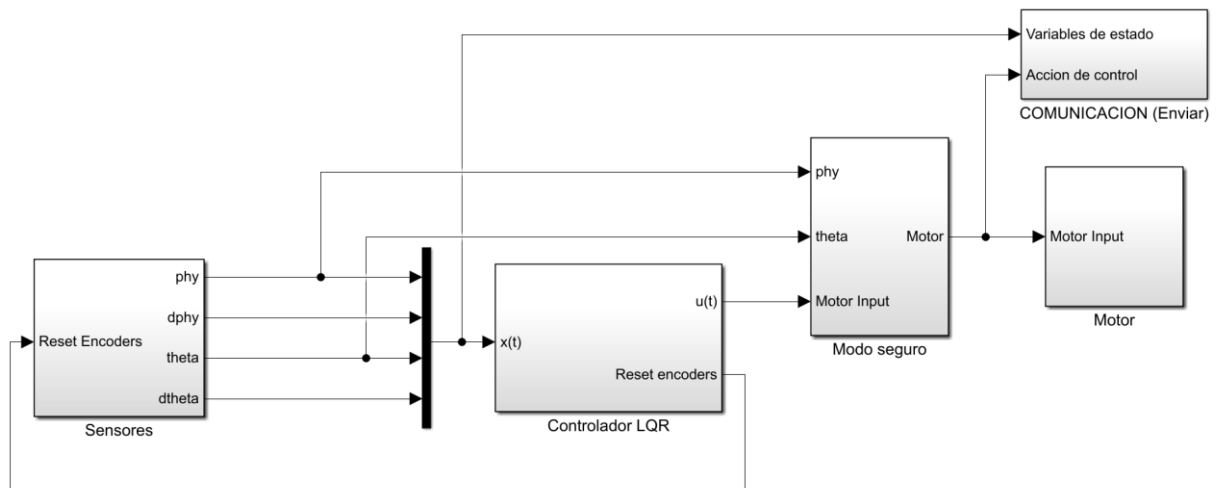


Ilustración 24. Modelo de Simulink ejecutado por el LEGO EV3

El primer subsistema se puede observar en la Ilustración 25, este subsistema se utiliza para leer los encoders, además de resetearlos. Está formado por tres subsistemas que se explican a continuación.

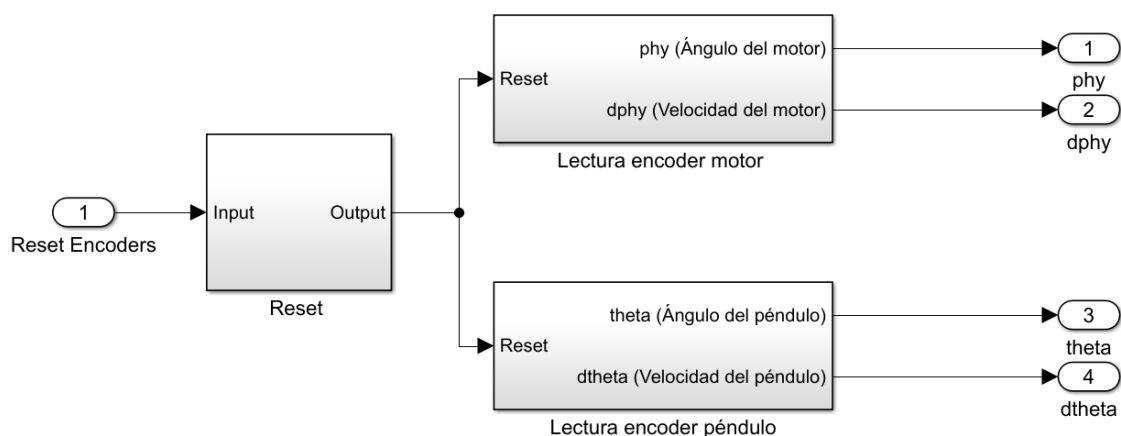


Ilustración 25. Subsistema Sensores

MEMORIA

El primer subsistema que forma parte del subsistema de la Ilustración 25, se muestra en la Ilustración 26. Sirve para realizar un reseteo de los sensores que miden los ángulos del motor y del péndulo. Existen tres opciones diferentes para resetear los encoders: al cargar el modelo en el LEGO EV3, mediante una señal de entrada recibida desde el ordenador o pulsando el botón central del ladrillo EV3. Esta señal se convertirá mediante el bloque 'Data Type Conversion' a una señal de tipo *int8* para poder ser procesada por el LEGO EV3.

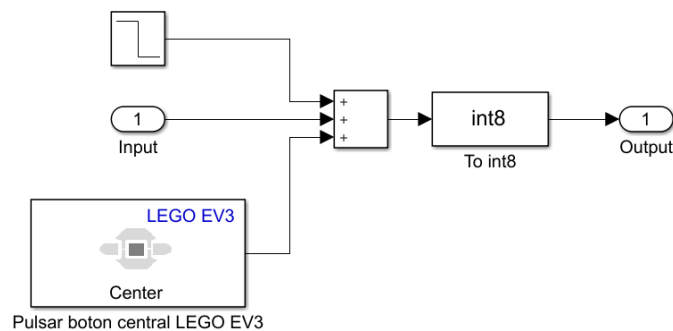


Ilustración 26. Subsistema Sensores -> Reset

En la Ilustración 27 se puede ver el segundo subsistema que se observa en la Ilustración 25. Sirve para leer el encoder que mide el ángulo de desplazamiento del motor y para calcular la velocidad angular del motor. Para ello se utiliza el bloque 'Encoder', que devuelve un valor en grados. Para usar la variable con mejor precisión en los posteriores cálculos se usa el bloque 'Data Type Conversion' para convertir la señal en tipo *double*. Después se usa un bloque 'Gain' para convertir los grados en radianes. Por último se utiliza un bloque 'Discrete State-Space' para derivar el ángulo y obtener la velocidad del motor. La entrada al subsistema es la salida de *reset* del anterior subsistema.

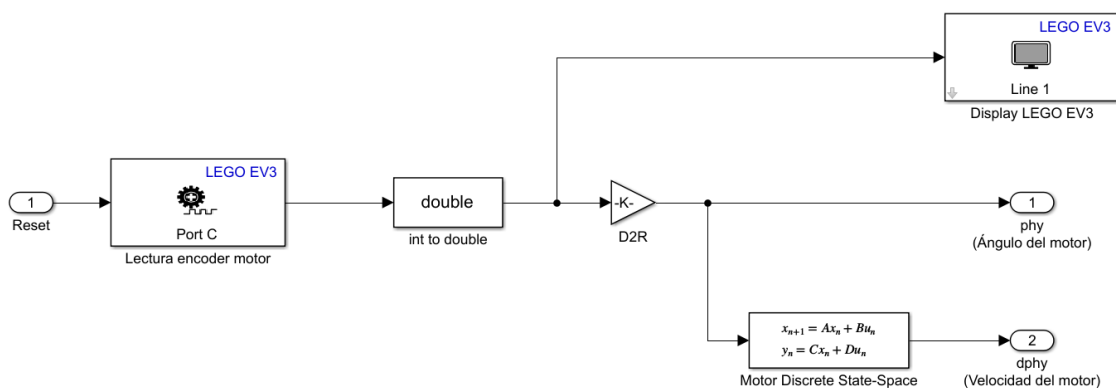


Ilustración 27. Subsistema Sensores -> Lectura encoder Motor

MEMORIA

En la Ilustración 28 se observa el tercer y último subsistema de la Ilustración 25. Actúa de la misma forma que el anterior subsistema pero para el encoder del péndulo. Sigue el mismo procedimiento que el subsistema de la Ilustración 27 con la única variación de que se utiliza un bloque 'Gain' para invertir el ángulo y la velocidad del péndulo.

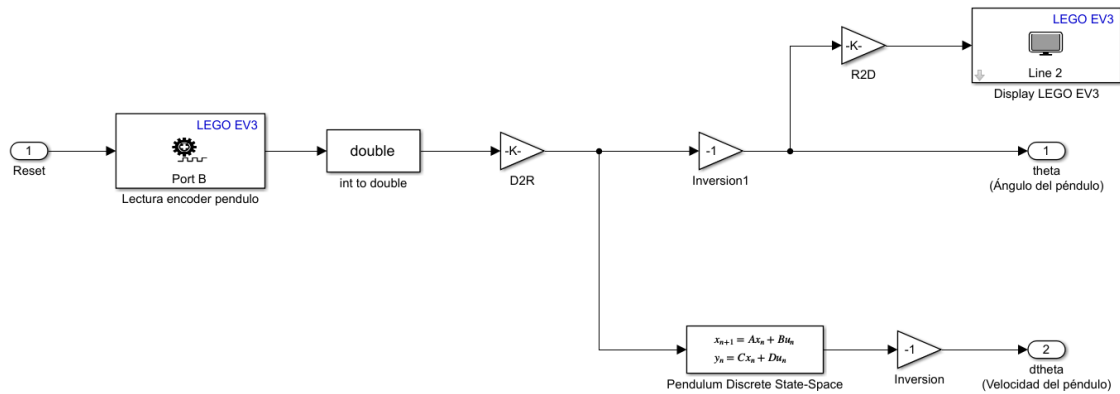


Ilustración 28. Subsistema Sensores -> Leer encoder Pendulo

MEMORIA

El subsistema Controlador LQR se puede ver en la Ilustración 29, es el encargado de implementar el control del péndulo. Además dentro de él se encuentra otro subsistema que es el encargado de recibir los datos del ordenador, este se observa en la Ilustración 30. La entrada a este subsistema representa los valores medidos por los sensores, que son las variables de estado del sistema. Tiene dos salidas: la acción de control para el motor y la señal recibida desde el ordenador para resetear los encoders. Existen cuatros bloques 'Product' para realizar el control por realimentación por variables de estado y un bloque 'Gain' para implementar la inversión de la ley de control. Por último un bloque 'Switch' para apagar o encender el motor según sea la señal recibida desde el ordenador (con un 1 pasa la acción de control, con un 0 el motor queda apagado).

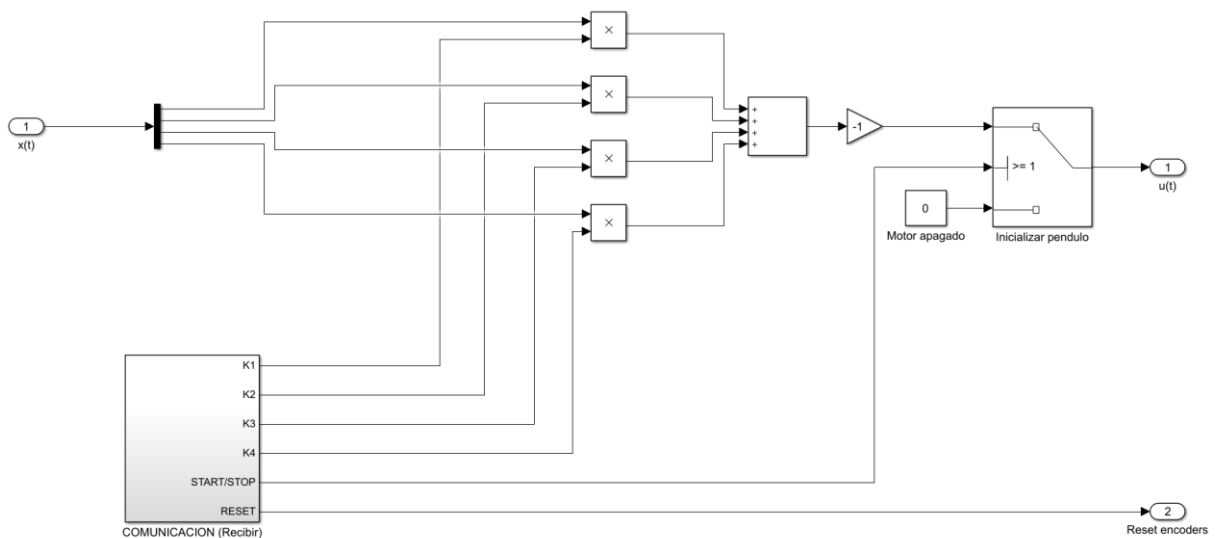


Ilustración 29. Subsistema Controlador LQR

Este subsistema es el encargado de recibir los datos desde el ordenador con un bloque 'UDP Receive'.

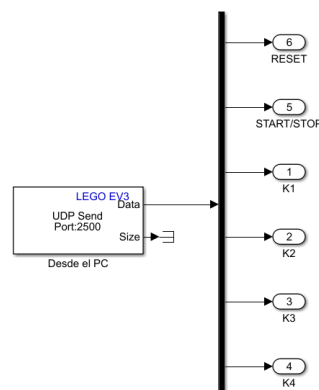


Ilustración 30. Subsistema Controlador LQR ->Comunicación (Recibir)

MEMORIA

Otro de los subsistemas que forman parte del modelo principal se muestra en la Ilustración 31, este subsistema se encarga de apagar el motor en caso de que se superen unas condiciones angulares implementadas en el subsistema de la Ilustración 32, para que los cables de los sensores no sufran a la hora de controlar el péndulo. Para ello si el ángulo del motor es superior a 360° o el péndulo no se encuentra en la zona equilibrio el motor se apagará.

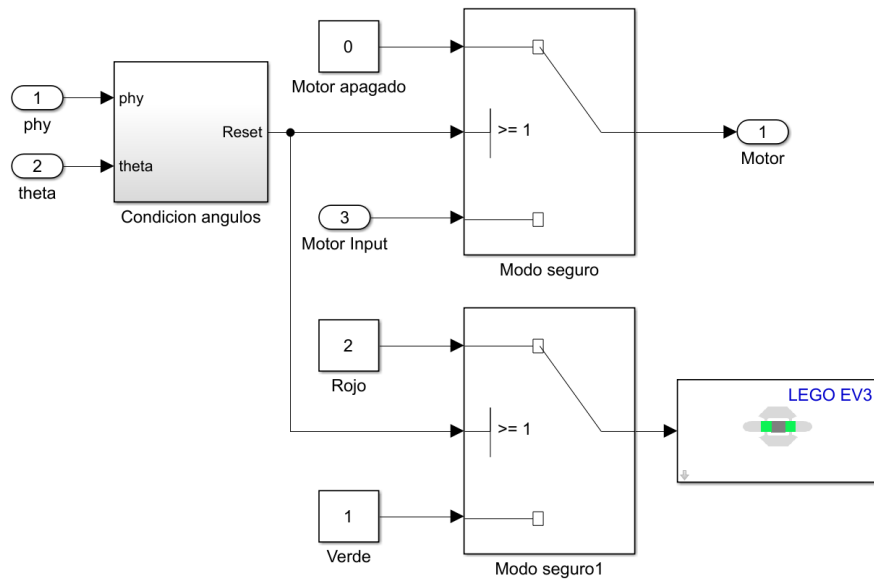


Ilustración 31. Subsistema Modo seguro

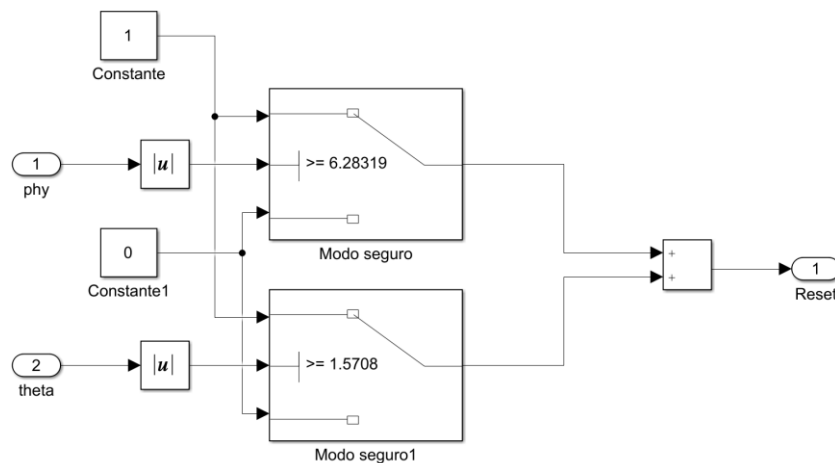


Ilustración 32. Subsistema Modo seguro -> Condición ángulos

MEMORIA

El subsistema encargado de enviar los datos al ordenador se muestra en la Ilustración 33. Al ordenador se envían los datos de las variables de estado, de la acción de control y además se introduce un reloj digital que envíe los datos de tiempo real con un intervalo de 0,01 segundos. Para ello se utiliza el bloque 'UDP Send'

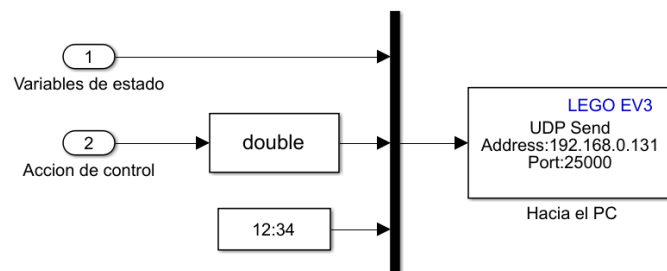


Ilustración 33. Subsistema Comunicación (Enviar)

Por último en la Ilustración 34, se puede ver el subsistema encargado de alimentar el motor LEGO EV3 Power Functions XL. Para ello se utiliza un bloque 'Motor' y un bloque 'Saturation' para limitar la acción de control entre los valores admisibles por el motor.

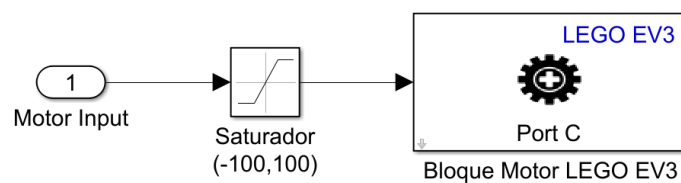


Ilustración 34. Subsistema Motor

MEMORIA

En la Ilustración 35 se puede observar el modelo de *Simulink* que se va a ejecutar en el ordenador. En el área CONTROL LEGO EV3 se observan los bloques 'Slider Switch', 'Push Button' y cuatro bloques 'Edit', estos bloques están asociados a los bloques 'Constant' de la Ilustración 36. Ellos determinan las variables de control que se enviarán a través del bloque 'UDP Send' al LEGO EV3. Por otro lado en el área COMUNICACIÓN LEGO EV3 se presentan los datos recibidos desde el LEGO EV3 a través del bloque 'UDP Receive'.

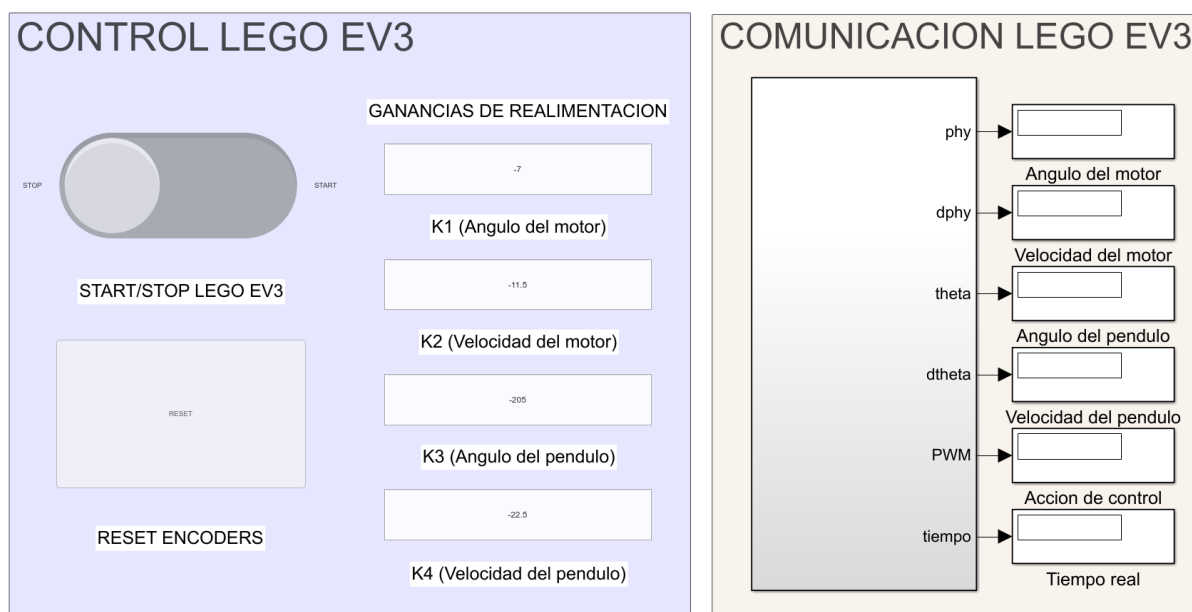


Ilustración 35. Modelo de Simulink ejecutado en el ordenador

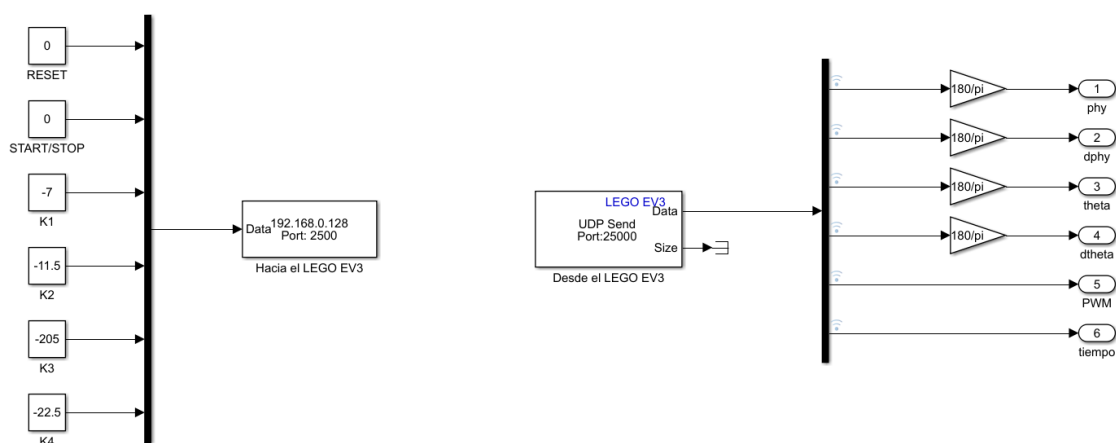


Ilustración 36. Comunicación ordenador - LEGO EV3

MEMORIA

Por último para poder ejecutar ambos modelos (el del PC y el del LEGO EV3) es necesario modificar los parámetros de ejecución. Modificando la pestaña **Solver** de *Simulink* como se observa en la Ilustración 37. El tiempo de muestro debe establecerse como *ts*, que será una variable establecida en el **Workspace** de *Matlabd* como *ts=0.01* (0.01 segundos de tiempo de muestreo).

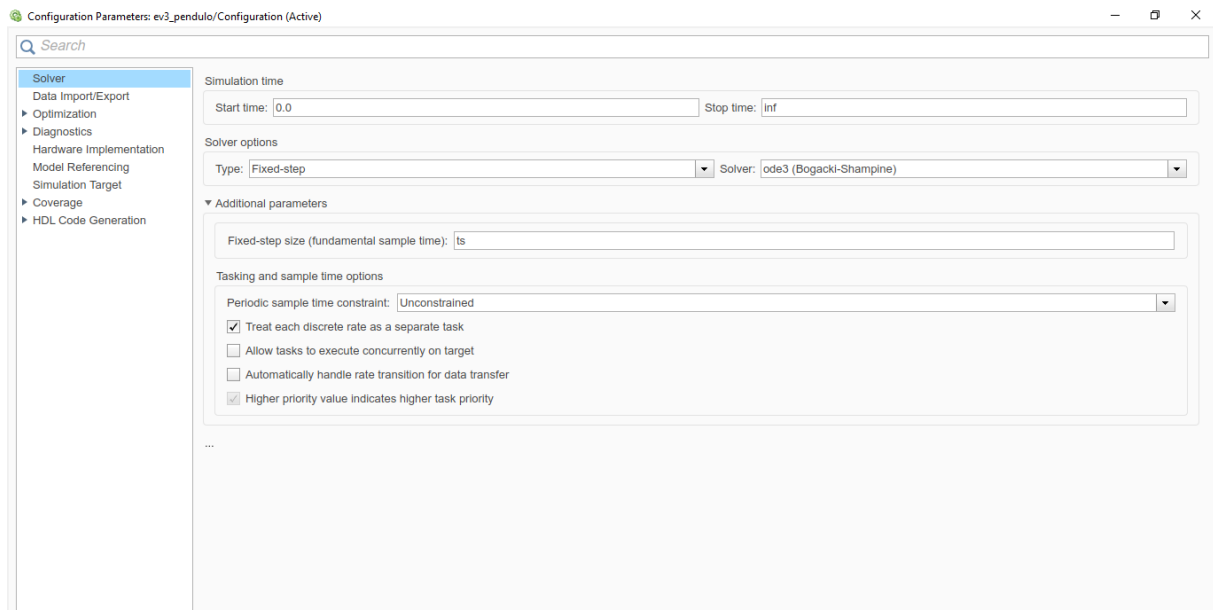


Ilustración 37. Parámetros del modelo de Simulink

Para el modelo a ejecutar en el LEGO EV3 habrá que modificar más parámetros que se detallan en el Anexo 2. Para ejecutar este modelo basta con pulsar el botón **Deploy to Hardware** en **Modo External**. Una vez ejecutado el modelo en el ladrillo EV3, pulsando el botón **Run** en **Modo Normal** en el modelo a ejecutar en el ordenador, se ejecutará dicho modelo en modo de simulación y ya se podrá controlar desde el ordenador el LEGO EV3.

Comportamiento del péndulo rotario invertido real

Utilizando el último controlador diseñado (Controlador 6), es el controlador que mejor respuesta ofrece en simulación, se realizan diversos experimentos sobre el péndulo. Las señales que representan las variables de estado y la acción de control se monitorizarán como se observa en la Ilustración 38. Las señales que se quieran recoger basta con seleccionarlás, pulsar el botón derecho y pulsar **Log Selected Signals**. Todas las señales que aparezcan con ese icono (se observan en los círculos amarillos de la Ilustración 38) aparecerán recogidas después de la simulación en el **Simulation Data Inspector** (círculo rojo de la Ilustración 38).

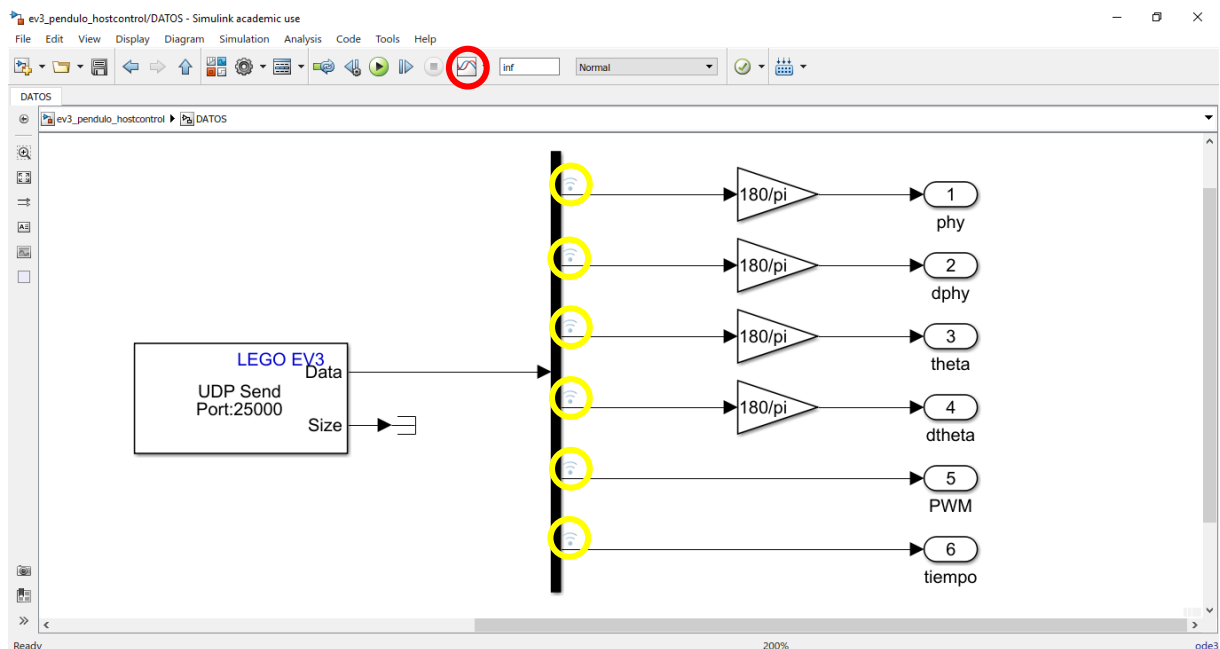


Ilustración 38. Señales del péndulo

En la Ilustración 39 se puede ver la ventana del **Simulink Data Inspector** que es la plataforma a través de la cual se van a obtener los datos reales del péndulo. Remarcado en rojo se observan los datos obtenidos en todas las simulaciones que se han realizado en el péndulo real.

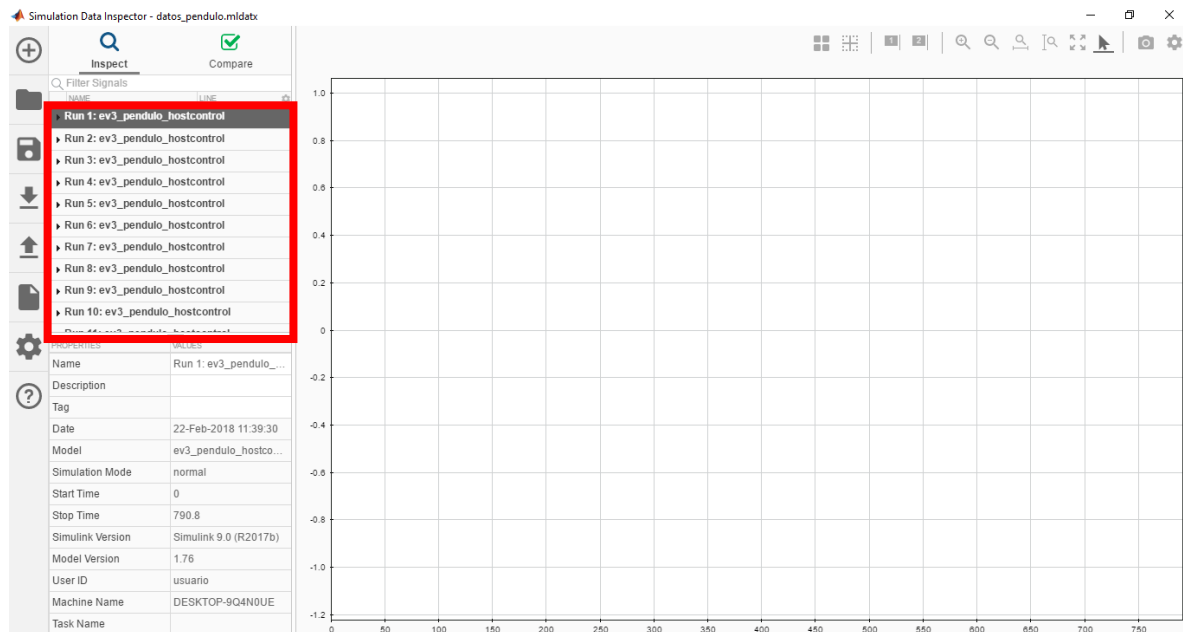


Ilustración 39. Interface Simulink Data Inspector

En la Ilustración 40 se muestran las *Log Selected Signals* que son respectivamente: ángulo del motor, velocidad angular del motor, ángulo del péndulo, velocidad angular del péndulo, acción de control y tiempo real transcurrido en el modelo ejecutado en el péndulo. Los datos obtenidos se deben exportar a *Matlab*, para poder procesarlos de manera adecuada, ya que el tiempo transcurrido en el modelo ejecutado del ordenador es mayor al real, ya que se trata de una simulación (*Modo Normal*) y no una ejecución en tiempo real (*Modo external*). Por ello el tiempo que recoge el *Simulation Data Inspector* no coincide con el tiempo real.

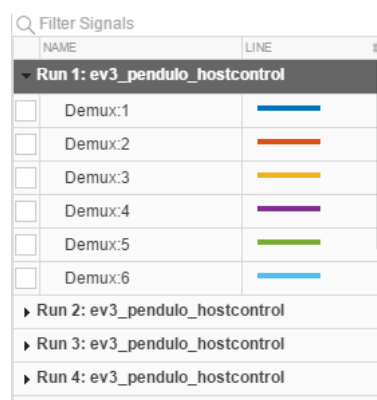


Ilustración 40. Log Selected Signals

MEMORIA

Utilizando un script como el siguiente es posible obtener las gráficas del comportamiento del péndulo a través de los datos recogidos en el *Simulink Data Inspector*. Para tener los datos reales del LEGO EV3 es necesario utilizar la instrucción *resample* para remuestrear los datos respecto al tiempo real obtenido desde el péndulo con un tiempo de muestro de 0,01 segundos (frecuencia de 100 Hz).

```
t = datos_t.Data;
phy = datos_phy.Data;
dphy = datos_dphy.Data;
theta = datos_theta.Data;
dtheta = datos_dtheta.Data;
u = datos_u.Data;
[phy1,t1]=resample(phy,t,100);
[dphy1,t1]=resample(dphy,t,100);
[thetal,t1]=resample(theta,t,100);
[dthetal,t1]=resample(dtheta,t,100);
[u1,t1]=resample(u,t,100);

subplot(3,1,1); plot(t1,phy1);
title('Angulo del motor')
xlabel('Tiempo (s)')
ylabel('rad')
subplot(3,1,2); plot(t1,thetal);
title('Angulo del pendulo')
xlabel('Tiempo (s)')
ylabel('rad')
subplot(3,1,3); plot(t1,u1);
title('Accion de control')
xlabel('Tiempo (s)')
ylabel('PWM')
```

A continuación se muestran las gráficas de los cinco experimentos realizados sobre el péndulo. En ellos se comprueba cinco veces como se realiza el control. En todos ellos el comportamiento del péndulo es muy similar.

Experimento 1

En este primer experimento se puede observar como se produce el control del péndulo en el punto de equilibrio a partir del segundo 20 hasta el segundo 50, que es el instante en el que desde el ordenador se para el motor del péndulo y se observa como el péndulo cae hasta estabilizarse en π radianes. Antes del segundo 20 lo único que se puede ver es como se sube el péndulo hasta la posición de equilibrio y como se resetean los encoders para que tanto el ángulo del motor, como el del péndulo sean igual a cero.

En el intervalo de 30 segundos en el que el péndulo invertido rotatorio se sitúa en el punto de equilibrio se observa:

- El ángulo del motor oscila entre -1 radián y 1 radián para asegurar que el péndulo se mantenga en el punto de equilibrio.
- El ángulo del péndulo se mantiene en torno a 0 radianes con pequeñas oscilaciones.
- La acción de control varía entre -100 y 100, asegurando el giro del motor para la estabilizar el péndulo.

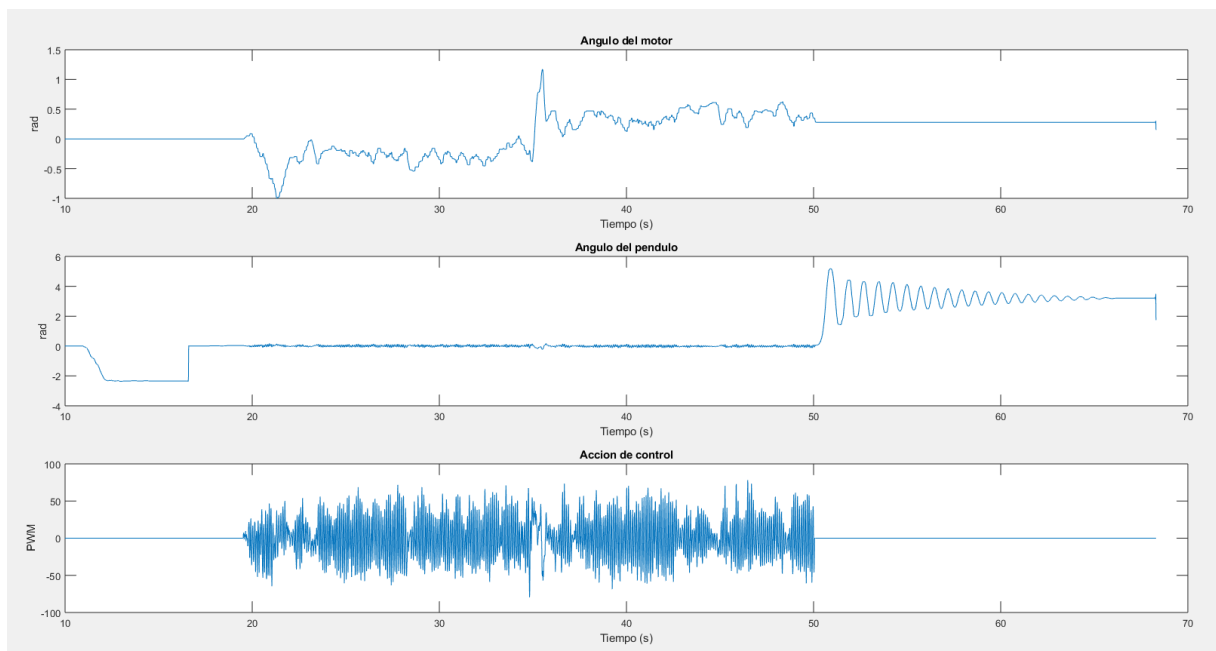


Ilustración 41. Experimento 1

Experimento 2

En este segundo experimento se puede observar como se produce el control del péndulo en el punto de equilibrio antes del segundo 20 hasta algo más tarde del segundo 50, que es el instante en el que desde el ordenador se para el motor del péndulo y observa como el péndulo cae hasta estabilizarse en π radianes. Antes del segundo 20 lo único que se puede ver es como se sube el péndulo hasta la posición de equilibrio y como se resetean los encoders para que tanto el ángulo del motor, como el del péndulo sean igual a cero.

En el intervalo de más 30 segundos en el que el péndulo invertido rotatorio se sitúa en el punto de equilibrio se observa:

- El ángulo del motor supera el ángulo del anterior experimento para asegurar que el péndulo se mantenga en el punto de equilibrio.
- El ángulo del péndulo se mantiene en torno a 0 radianes con pequeñas oscilaciones, al igual que en el anterior experimento.
- La acción de control se comporta de manera similar al anterior experimento.

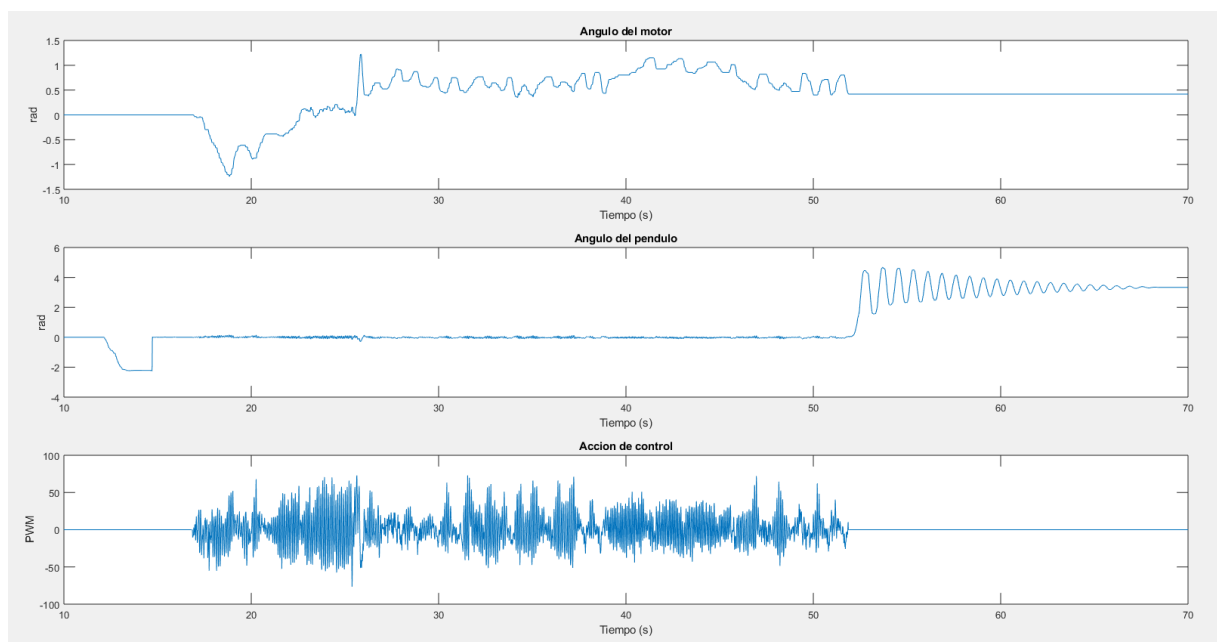


Ilustración 42. Experimento 2

Experimento 3

En este tercer experimento se puede observar como se produce el control del péndulo en el punto de equilibrio de la misma forma que en los anteriores experimentos.

En el intervalo de más 30 segundos en el que el péndulo invertido rotatorio se sitúa en el punto de equilibrio se observa:

- El ángulo del motor se estabiliza en torno a -2 radianes para asegurar que el péndulo se mantenga en la posición de equilibrio.
- El ángulo del péndulo se mantiene en torno a 0 radianes con pequeñas oscilaciones, al igual que en los anteriores experimentos.
- La acción de control en este caso se sitúa entre -50 y 50.

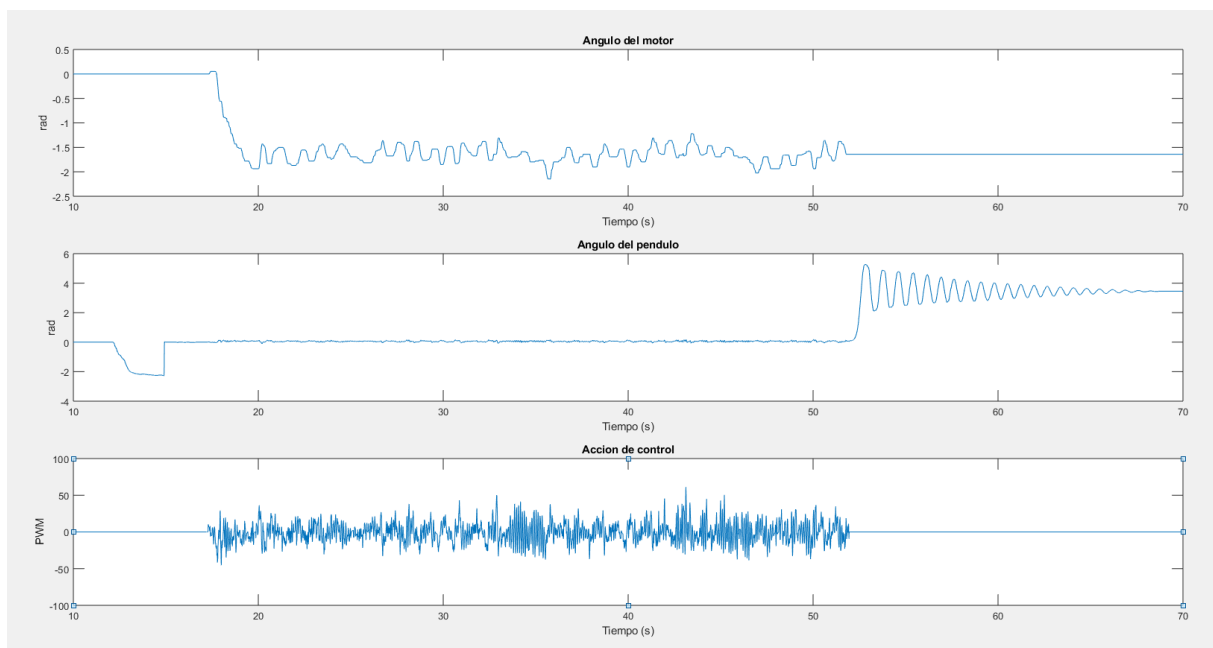


Ilustración 43. Experimento 3

Experimento 4

En este cuarto experimento, a diferencia de los anteriores, se puede observar como se produce el control del péndulo hasta algo más tarde del segundo 30, donde ya no es posible controlar el péndulo y éste cae hasta estabilizarse en $-\pi$ radianes.

En el intervalo de unos 15 segundos en el que el péndulo péndulo invertido rotatorio se sitúa en el punto de equilibrio se observa:

- El ángulo del motor se mantiene en torno a 0 radianes, para terminar girando casi 2π radianes, para poder mantener el péndulo en el punto de equilibrio.
- El ángulo del péndulo se mantiene en torno a 0 radianes con pequeñas oscilaciones, pero pasado el segundo 30, cae a $-\pi$ radianes.
- La acción de control en el momento en el que el péndulo se cae supera la barrera de -100 y 100, para intentar estabilizar el péndulo.

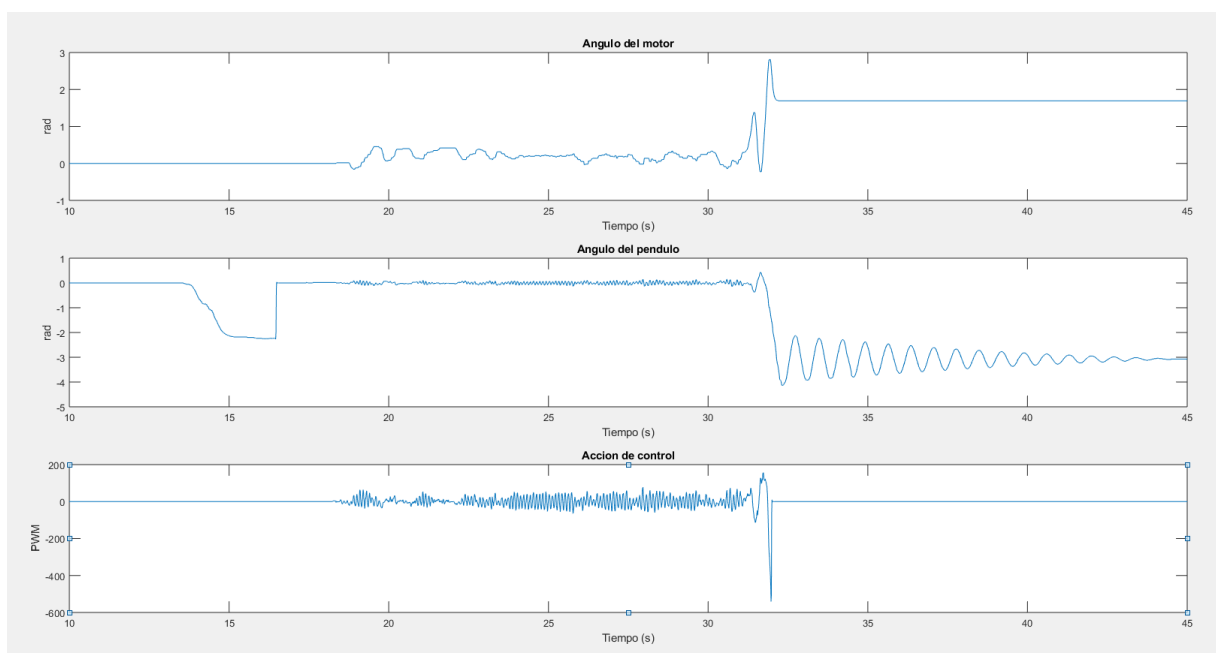


Ilustración 44. Experimento 4

Experimento 5

Por último en este quinto experimento se observa lo mismo que los tres primeros. Con la única diferencia de al finalizar la acción del motor desde el ordenador, el péndulo cae hasta estabilizarse en $-\pi$ radianes.

En el intervalo de tiempo en el que el péndulo invertido rotatorio se sitúa en el punto de equilibrio se observa:

- El ángulo del motor se mantiene en torno a -1 radián para poder mantener el péndulo en el punto de equilibrio.
- El ángulo del péndulo se mantiene en torno a 0 radianes con pequeñas oscilaciones.
- La acción de control no supera los valores de -100 y 100.

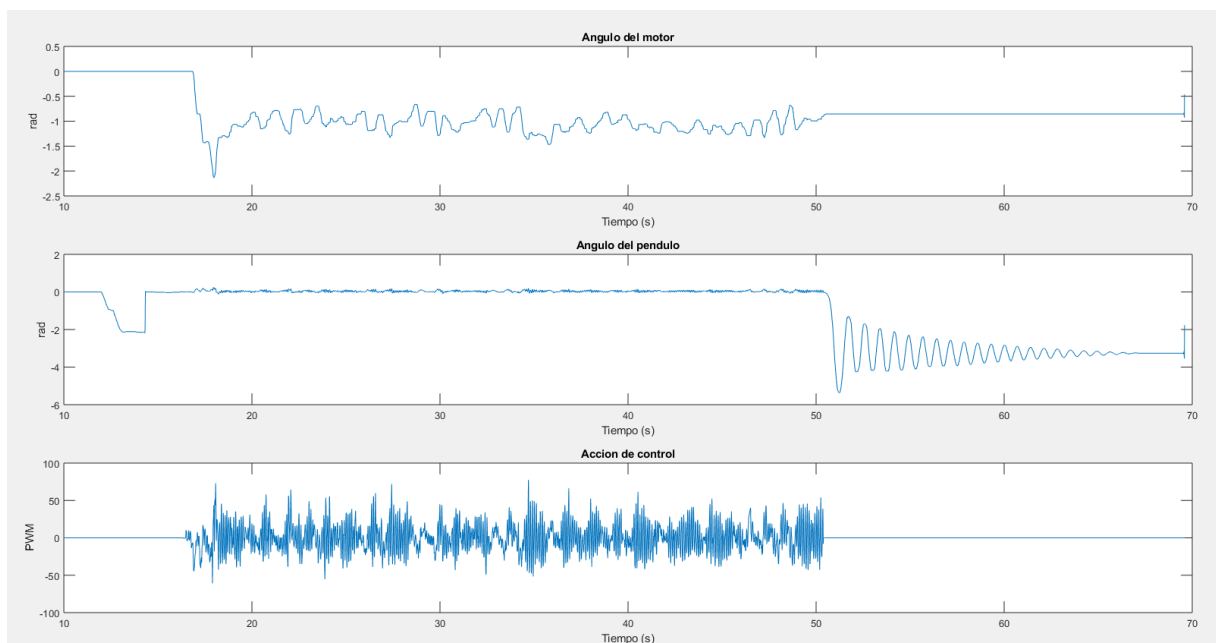


Ilustración 45. Experimento 5

8. Conclusiones

En la primera parte del trabajo, se ha realizado un modelo dinámico del péndulo invertido, que se ha implementado utilizando Matlab y Simulink para su simulación. Para obtener las ecuaciones que describen el comportamiento del sistema se han utilizado las ecuaciones diferenciales del movimiento de *Euler-Lagrange*. Cabe destacar que el modelo que se ha obtenido ha presentado carácter no lineal.

Una vez modelado el comportamiento del sistema, se ha desarrollado un modelo lineal de pequeña señal en torno al punto de equilibrio sobre el que se desea gobernar el péndulo. Dicho modelo ha sido desarrollado en el espacio de estados. Se ha optado por este modo de representación del comportamiento del sistema por considerarse más adecuado para gobernar un sistema sub-actuado.

El control del sistema se ha efectuado empleando el método de control por realimentación de variables de estado. La finalidad del algoritmo de control ha sido mantener el brazo vertical del péndulo en posición vertical hacia arriba, con oscilaciones mínimas y empleando incrementos en el ángulo del rotor del motor de corriente continua de amplitud mínima. Para la obtención de las ganancias de realimentación de estado se ha utilizado la metodología de control óptimo (*LQR*).

En la segunda parte del trabajo se han implementado los controladores sobre el péndulo invertido rotatorio que se ha construido. Para ello, se ha realizado una identificación y posterior validación de los parámetros que intervienen en el modelo matemático. Una vez han sido obtenidos los nuevos parámetros del sistema, se han reajustado las ganancias de realimentación y se ha implementado el lazo de control en el sistema real.

Para ello ha sido necesario utilizar *Matlab* y *Simulink* para desarrollar las herramientas de software. Con *Simulink* se ha desarrollado el sistema de control a cargar en el micro controlador del LEGO EV3, que lee los sensores, calcula la ley de control y la manda al actuador en tiempo real.

Además, se ha utilizado también *Simulink*, para desarrollar una interface para el usuario. Esta interface establece una comunicación con el péndulo y monitoriza los datos de movimiento y actuación del mismo. Esta interface permite al usuario mandar comandos al péndulo para: arrancar o parar la actuación del motor, resetear los encoders o modificar el valor de las ganancias de realimentación. Con todos estos elementos se ha obtenido un conjunto de software que permite controlar en tiempo real el estado del sistema, y además, permite almacenar los datos generados durante los experimentos para su posterior análisis.



MEMORIA

La mayor dificultad que ha conllevado el desarrollo del trabajo ha sido el hardware usado por el LEGO EV3. La identificación de los parámetros que caracterizan el motor LEGO Power Functions XL ha sido muy compleja.

Este trabajo abre la puerta a la continuación del mismo en el desarrollo de nuevos controladores utilizando diferentes técnicas de control, las posibilidades que ofrece son enormes. Además este sistema no solo plantea el problema de mantenerlo estable en el punto de equilibrio, existe el trabajo adicional de diseñar un controlador para llevar el péndulo hasta la zona de equilibrio.

A nivel docente se convierte en la manera perfecta de poder mostrar el funcionamiento de sistemas no lineales y sub-actuados. Permite la introducción de los alumnos al control en el espacio de estados, aplicando técnicas de control que son completamente diferentes a las teorías de control convencionales.

“CONTROL DE UN PÉNDULO INVERTIDO ROTATORIO CON HARDWARE DE BAJO COSTE”



DOCUMENTO N° 3 ANEXOS

Peticionario:	Universidad de La Rioja
Informantes:	Fernando Chicote Navaridas
	Estudiante de Ingeniería Electrónica Industrial y Automática

ÍNDICE

ANEXO 1. MONTAJE DE LA MAQUETA	85
ANEXO 2. SOPORTE LEGO MINDSTORMS EV3 SIMULINK	97
1. Primeros pasos con LEGO MINDSTORMS EV3 Hardware	98
Primer paso.....	98
Segundo paso.....	98
2. Librería de bloques LEGO MINDSTORMS EV3	99
Bloque Encoder	100
Bloque Motor.....	101
Bloque UDP Send.....	102
Bloque UDP Receive	103
3. Comunicación con LEGO MINDSTORMS EV3 Hardware	104
ANEXO 3. MOTOR DE CORRIENTE CONTINUA	105
1. Modelo matemático de un motor de corriente continua	105
2. Obtención experimental de la planta	108
3. Características del motor.....	109

ANEXO 1. MONTAJE DE LA MAQUETA

En este anexo se va a detallar y explicar cómo se ha realizado el montaje del péndulo rotatorio invertido real, utilizando la plataforma LEGO Mindstorms EV3.

Para el montaje de la maqueta real del péndulo se han utilizado:

- 1 Kit LEGO MINDSTORMS Education EV3.
- 1 Set de Expansión EV3.
- 1 WNA1100: adaptador USB inalámbrico N150 (Wi-Fi Dongle).
- 1 Motor LEGO Power Functions XL (actuador).
- 2 GlideWheel-M Encoders (sensores).



Ilustración 46. Kit LEGO MINDSTORMS Education EV3



Ilustración 47. Set de Expansión EV3



Ilustración 48. Adaptador USB inalámbrico



Ilustración 49. Motor LEGO Power Functions XL

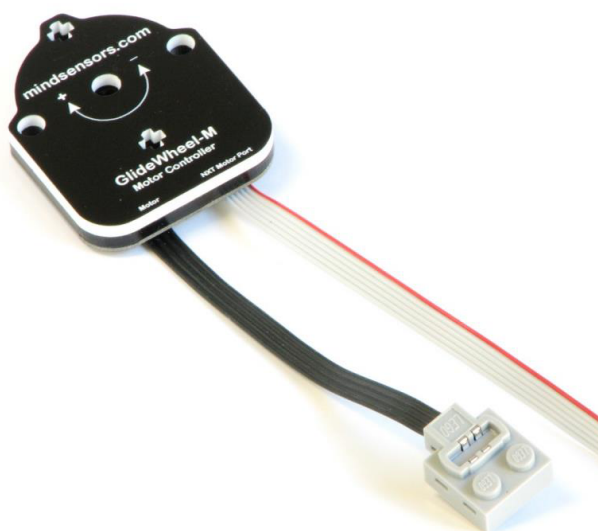


Ilustración 50. GlideWheel-M Encoders

ANEXOS

El proceso para montar el péndulo rotario invertido real se realiza de la siguiente forma:

En primer lugar, se construye la parte inferior de la estructura del péndulo. La parte inferior es en la que está situado el ladrillo EV3 y sostiene la estructura que alberga el motor y los dos encoders. El ladrillo irá anclado al lateral de la estructura. En la Ilustración 52 se puede observar la parte inferior ya montada.

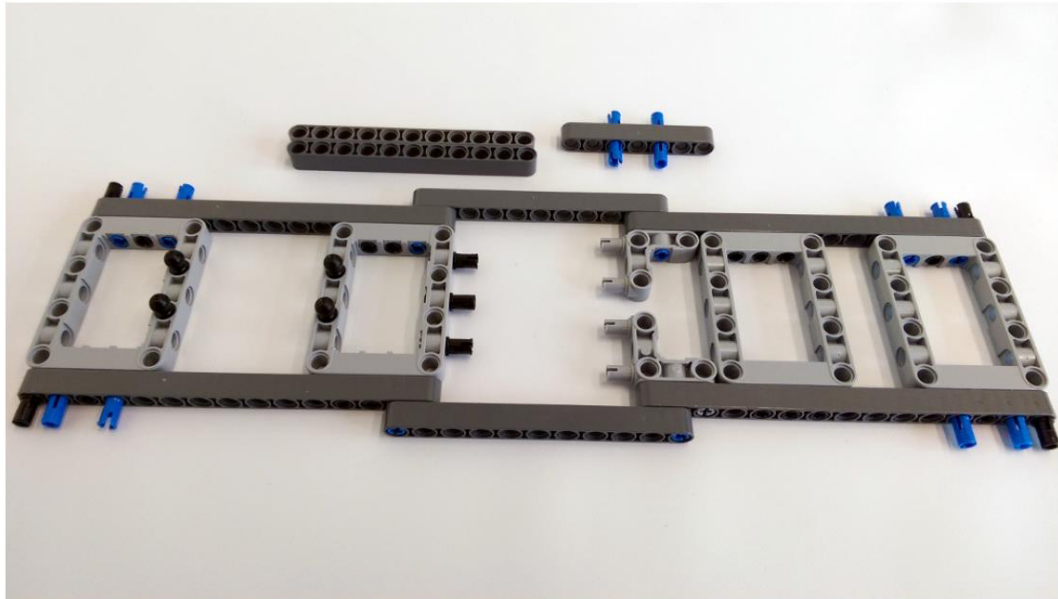


Ilustración 51. Parte superior de la estructura

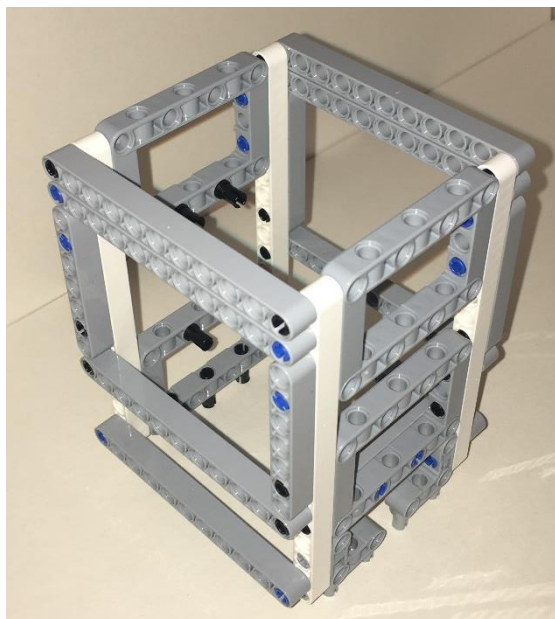


Ilustración 52. Parte superior de la estructura

ANEXOS

En la Ilustración 53 se pueden ver cuatro escuadras que se utilizan para otorgar mayor estabilidad a la parte inferior de la estructura. Las escuadras van ancladas en los laterales e impiden el movimiento lateral de la estructura.



Ilustración 53. Escuadras para la parte superior de la estructura

En la Ilustración 54 se muestra el ladrillo EV3 con los anclajes para integrarlo en el lateral de la parte inferior de la estructura. Además se muestra la base de la estructura total que se ve de forma más clara en la Ilustración 55. En la Ilustración 56 se observa la estructura inferior ya montada sobre la base, además el ladrillo EV3 ya se encuentra integrado en ella.

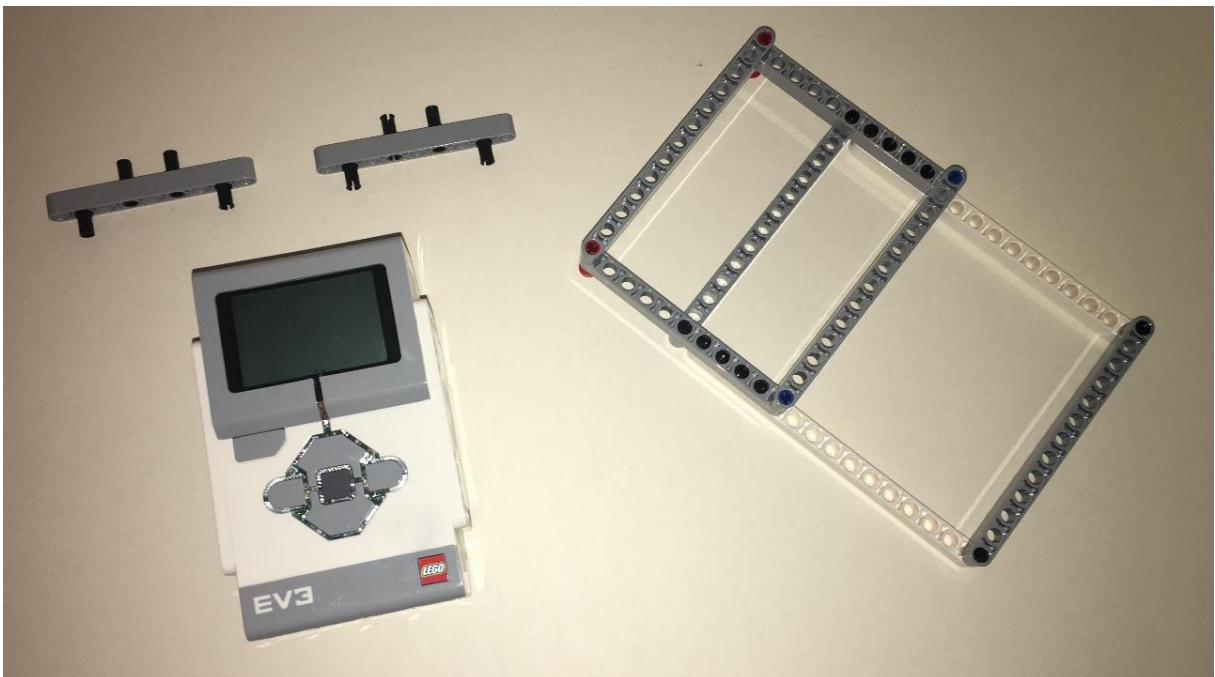


Ilustración 54. Anclajes para el ladrillo EV3 y base de la estructura

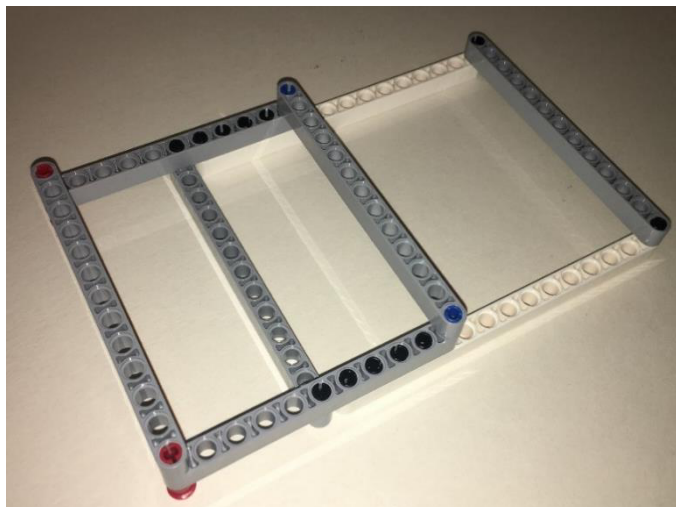


Ilustración 55. Base de la estructura

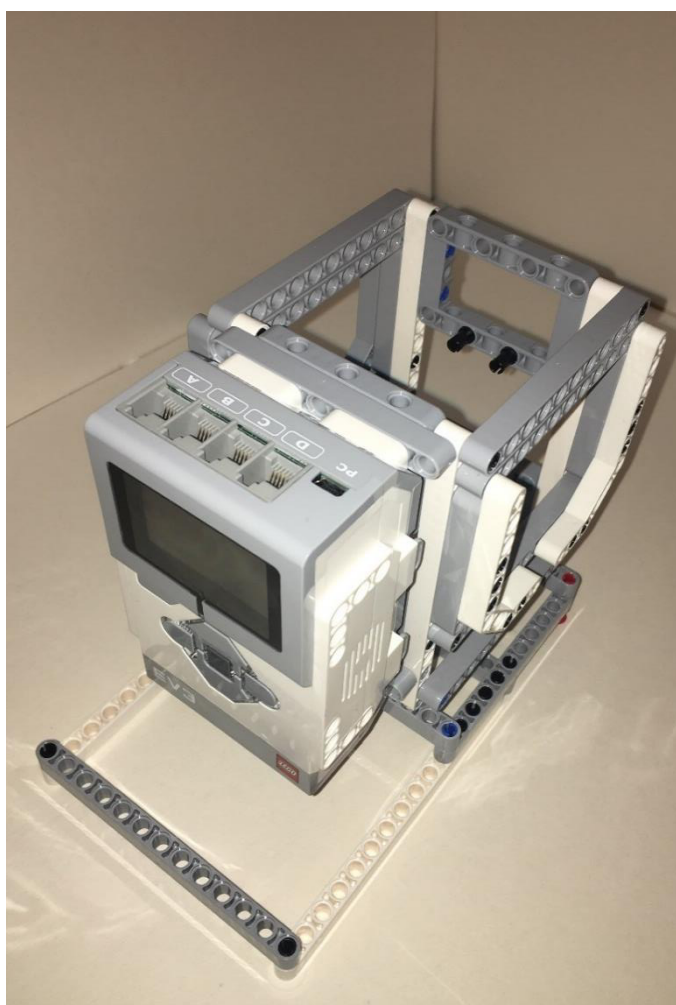


Ilustración 56. Parte inferior de la estructura

ANEXOS

Posteriormente, se monta la parte superior de la estructura del péndulo. La parte superior va anclada a la parte inferior. Además, esta parte sirve de sujeción para el motor del péndulo. Sobre ella se situará el péndulo. En la Ilustración 57 se muestran las diferentes piezas de LEGO que la componen.

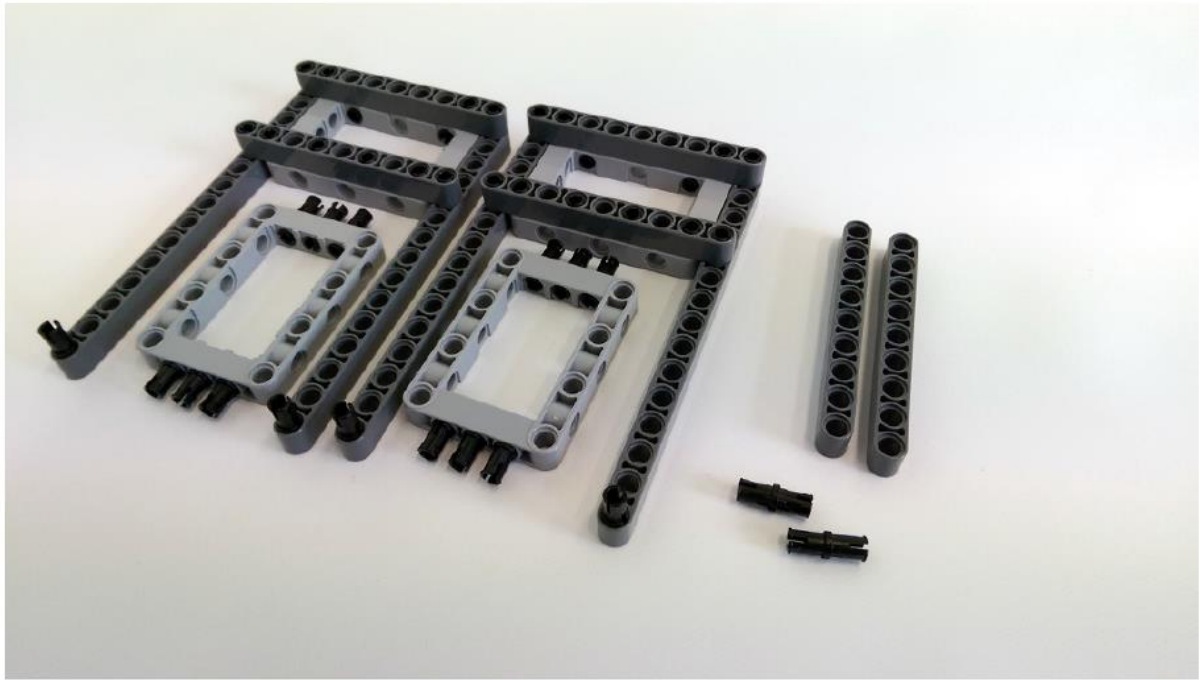


Ilustración 57. Parte superior de la estructura

ANEXOS

En la Ilustración 58 y en la Ilustración 59 se puede ver la estructura junto a la cual está montada el motor. Esta estructura permite anclar el motor a la parte superior anteriormente mencionada, y por otro lado permite enganchar el encoder que mide el ángulo del motor.

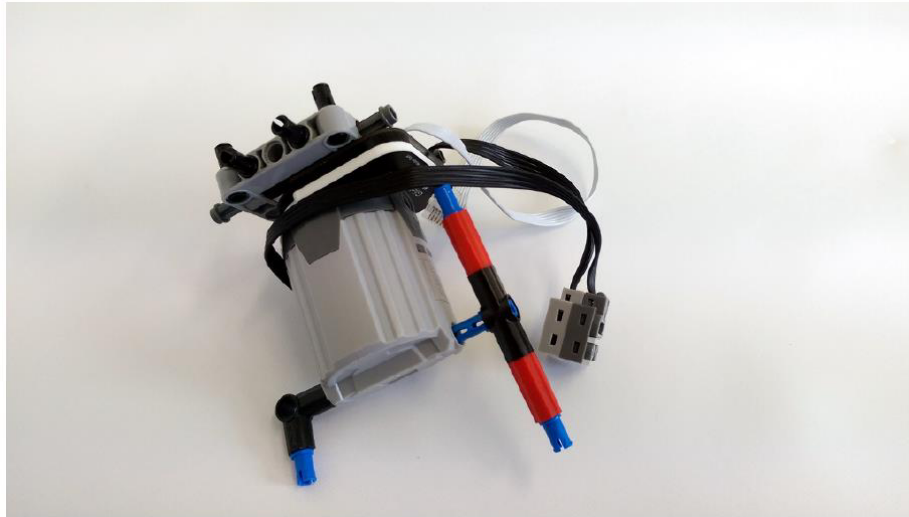


Ilustración 58. Montaje del motor

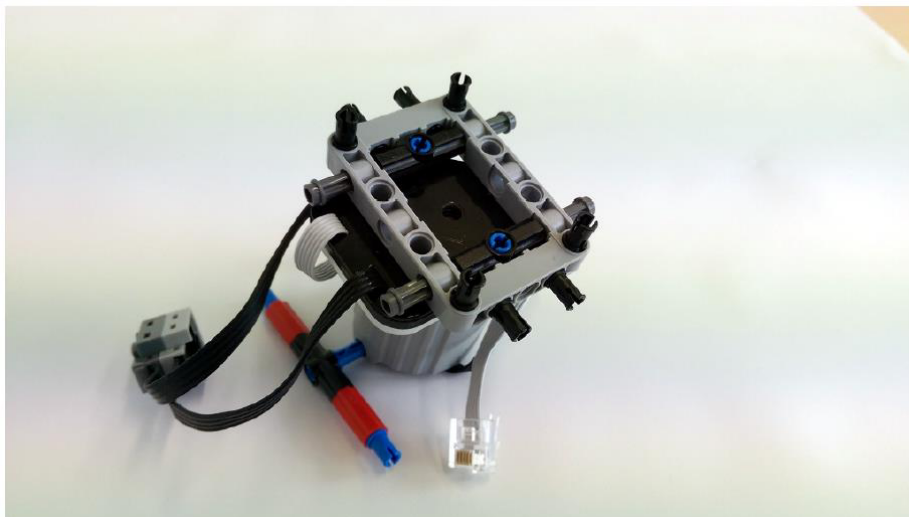


Ilustración 59. Montaje del motor

ANEXOS

En la Ilustración 60 se observa la forma en la que queda integrado el motor en la parte superior de la estructura, con dicha estructura ya montada. En la siguiente página, con la Ilustración 61 queda configurada la estructura total.

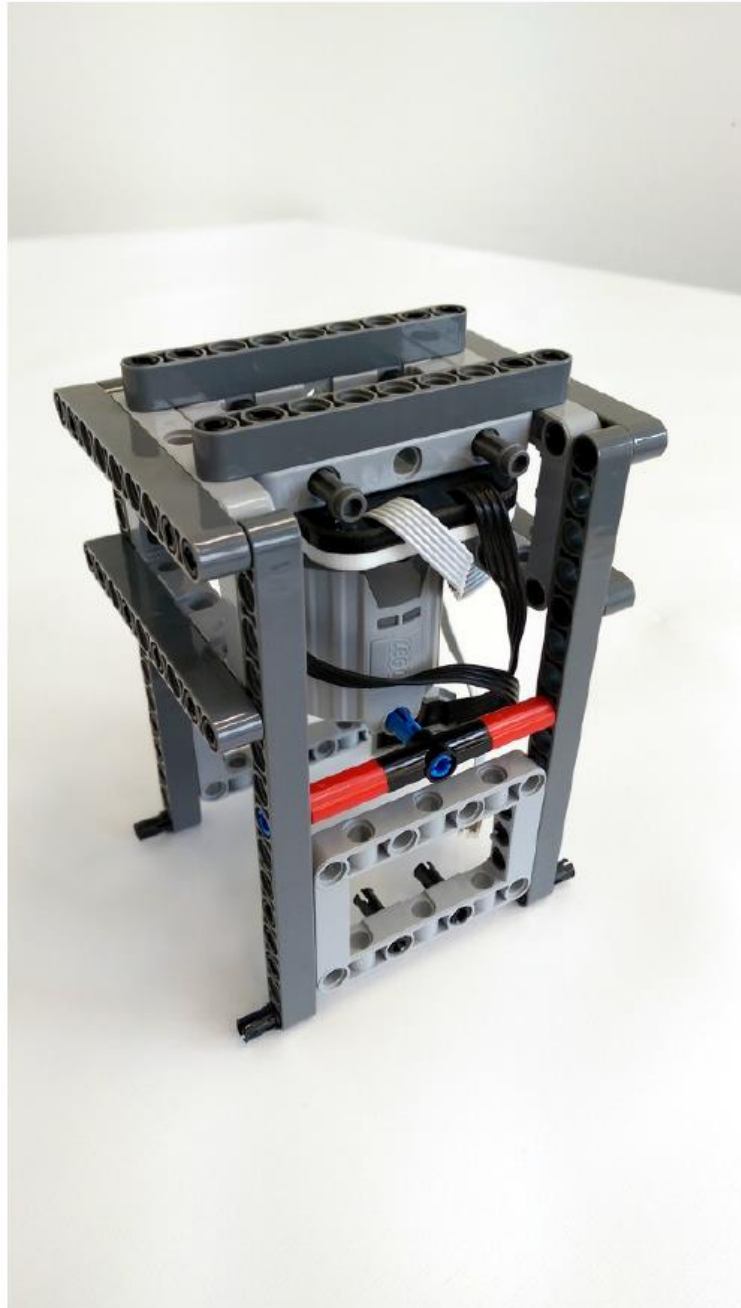


Ilustración 60. Motor integrado en la parte superior

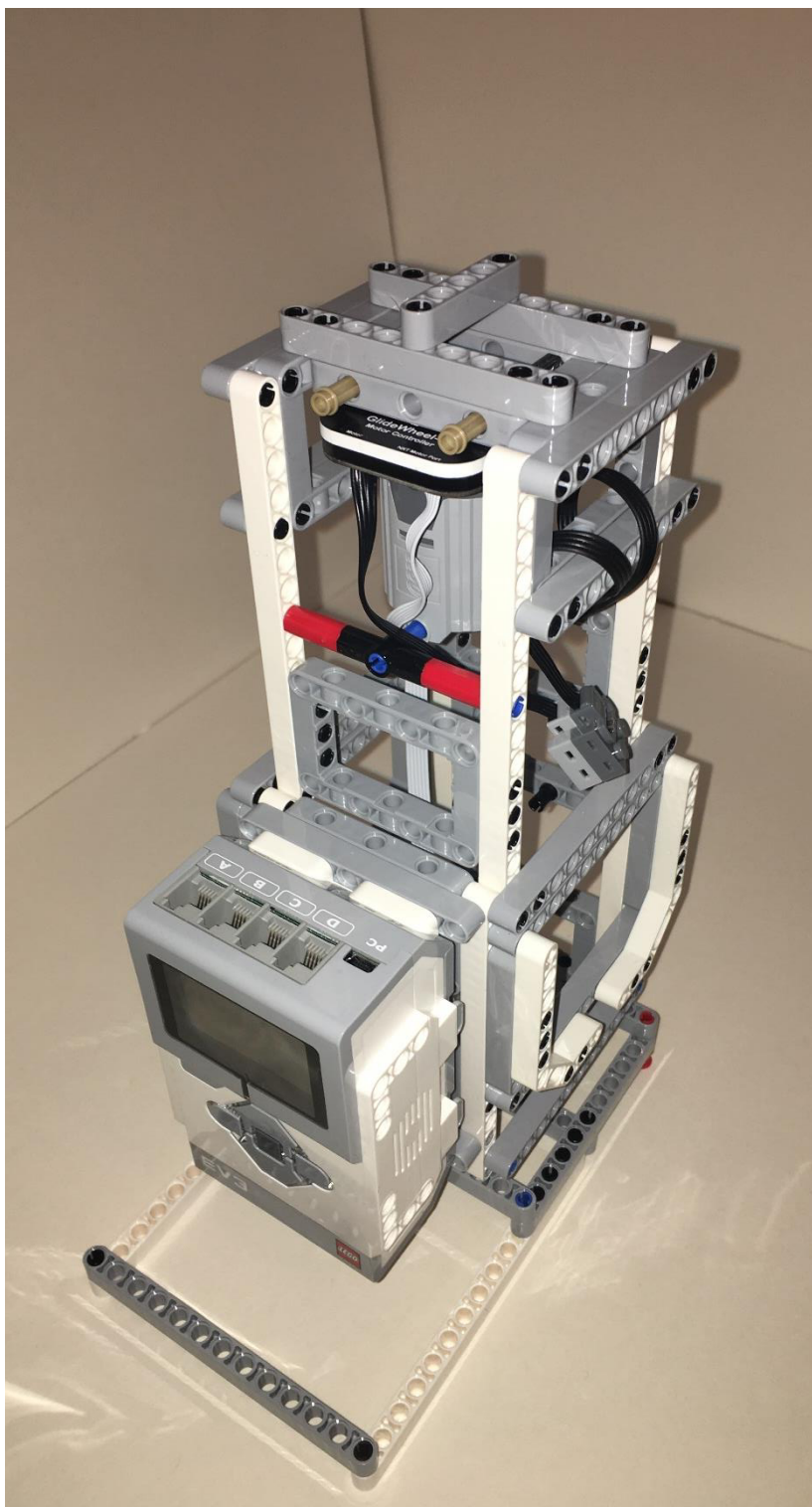


Ilustración 61. Estructura total

ANEXOS

En la Ilustración 62 y en la Ilustración 63 se muestran la configuración del péndulo. Se observan las piezas de LEGO que lo forman, junto con el encoder que medirá el ángulo del péndulo.



Ilustración 62. Montaje encoder y brazo del péndulo



Ilustración 63. Montaje encoder y brazo del péndulo

ANEXOS

Por último en la Ilustración 64 y en la Ilustración 65 se muestra la configuración final de toda la estructura que forma parte del péndulo rotatorio invertido.

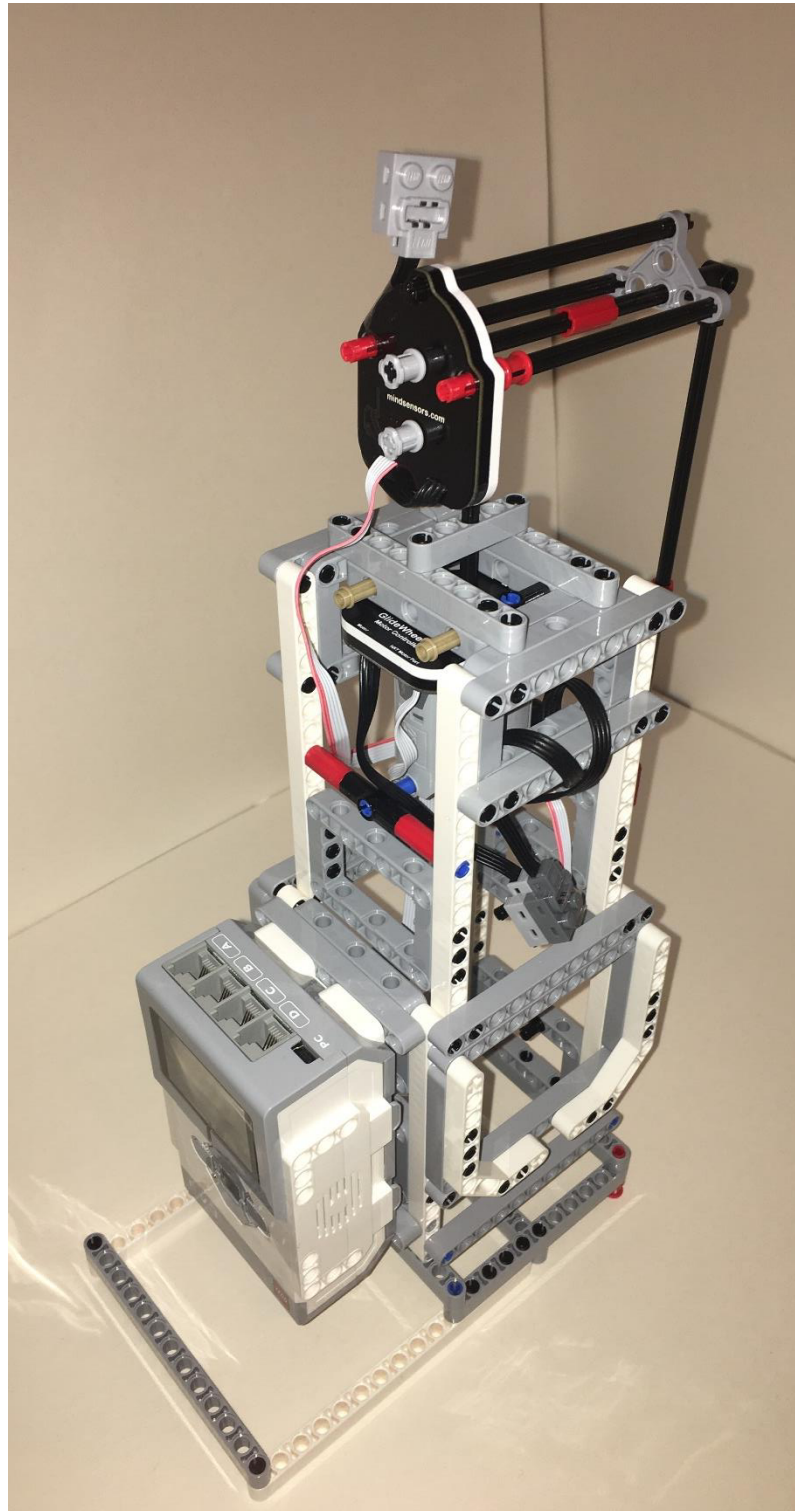


Ilustración 64. Estructura del péndulo invertido rotatorio

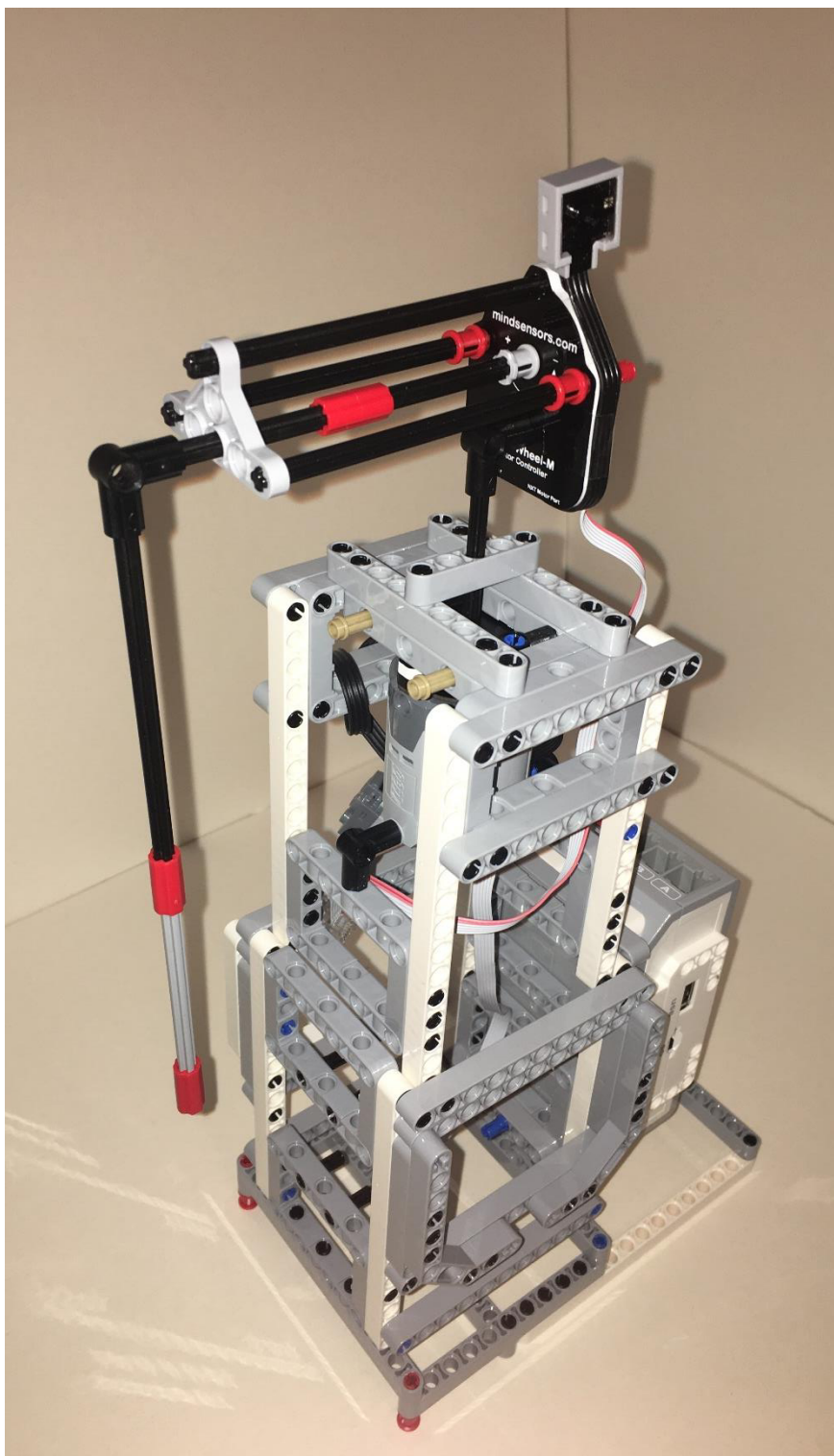


Ilustración 65. Estructura del péndulo invertido rotatorio

ANEXO 2. SOPORTE LEGO MINDSTORMS EV3

En este anexo se detalla cómo funciona el paquete adicional de *Simulink* para trabajar con la plataforma LEGO MINDSTORMS EV3.

El *Simulink Support Package* para el hardware LEGO MINDSTORMS EV3 permite programar y ejecutar algoritmos en los robots LEGO MINDSTORMS EV3. Este paquete de *Simulink* incluye una biblioteca de bloques para configurar y acceder a los sensores, actuadores e interfaces de comunicación del robot.

Es posible desarrollar modelos y algoritmos en *Simulink*, simularlos para comprobar que se ejecuten según lo previsto y cargar ese algoritmo completo en el dispositivo para su completa ejecución (a través de USB, Bluetooth, Wi-Fi o conexión Ethernet). También existe la posibilidad de poder ajustar parámetros en vivo desde el modelo de *Simulink* mientras el algoritmo se ejecute en el ladrillo EV3.

Este paquete incluye:

- Una biblioteca de bloques *Simulink* que se conectan a LEGO MINDSTORMS EV3, incluidos botones y altavoces; numerosos sensores de entrada y sensores de salida.
- Ajuste interactivo de parámetros de salidas y monitoreo de señal de entradas de sensores para aplicaciones que se ejecutan en LEGO MINDSTORMS EV3.

Para descargar este paquete basta con seguir las instrucciones que se detallan en esta página web:

<https://es.mathworks.com/help/supportpkg/legomindstormsev3/ug/install-target-for-lego-mindstorms-ev3-hardware.html>

1. Primeros pasos con LEGO MINDSTORMS EV3 Hardware

Primer paso

El primer paso para usar el ladrillo EV3 es confirmar su versión de firmware:

1. Encender el ladrillo EV3.
2. Navegar por el ladrillo EV3 hasta la pestaña **Settings > Brick Info**.
3. Verificar si la versión es **V1.03E** o superior.
4. Si no es así, utiliza el software de LEGO EV3 para actualizarlo.



Ilustración 66. Comprobación versión de firmware

Segundo paso

El segundo paso consiste en establecer una comunicación entre el ladrillo EV3 y el ordenador. Existen cuatro opciones diferentes, en este caso se va a realizar a través de la red inalámbrica, para ello:

1. Conectar el Wi-Fi Dongle en puerto USB del ladrillo EV3.
2. Navegar por el ladrillo EV3 hasta la pestaña **Settings > Brick Info**. Habilitar el WiFi, seleccionar una red y conectarse.
3. Navegar por el ladrillo EV3 hasta la pestaña **Settings > Brick Info**. Buscar el ID del dispositivo.
4. Navegar por el ladrillo EV3 hasta la pestaña **Settings > Brick Info**. Buscar la dirección IP del dispositivo.



Ilustración 67. Dirección IP del dispositivo

Con estos dos pasos ya estaría configurado el ladrillo EV3 para poder enviar cualquier modelo desde *Simulink*.

2. Librería de bloques LEGO MINDSTORMS EV3

En la Ilustración 68 se muestra la librería de bloques incluida en el paquete *Simulink Support Package* para el hardware LEGO MINDSTORMS EV3. Pulsando dos veces el botón izquierdo sobre cualquier bloque se obtiene una descripción de dicho bloque y los parámetros del mismo. Se va a prestar especial atención a los bloques: 'Encoder', 'Motor', 'UDP Send' y 'UDP Receive'; que son los de mayor interés para los modelos del péndulo.

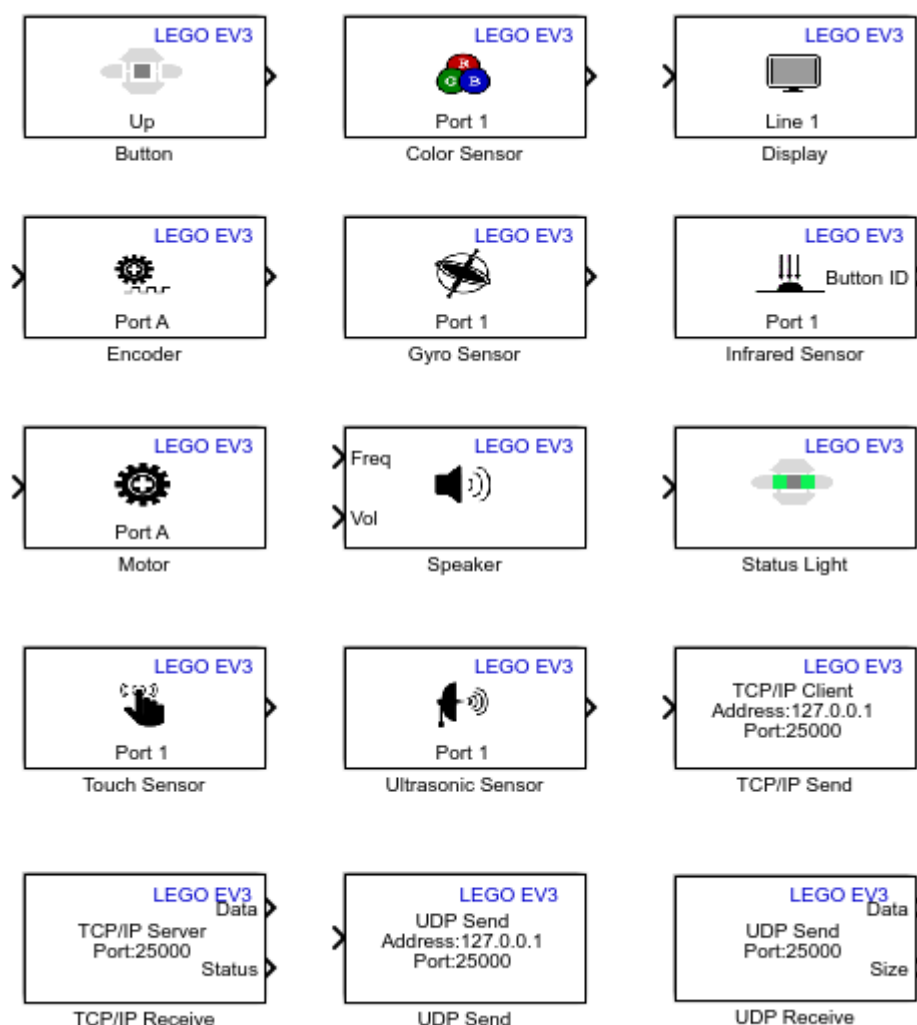


Ilustración 68. Librería LEGO MINDSTORMS EV3

Bloque Encoder

Este bloque mide la rotación acumulada de un motor para LEGO EV3 en grados. Suele estar asociado a un bloque motor. Puede resetearse cada muestra o mediante un señal externa. La salida del bloque emite datos del tipo *int32*. Otro parámetro de configuración es el puerto al que esta conectado en el ladrillo EV3.

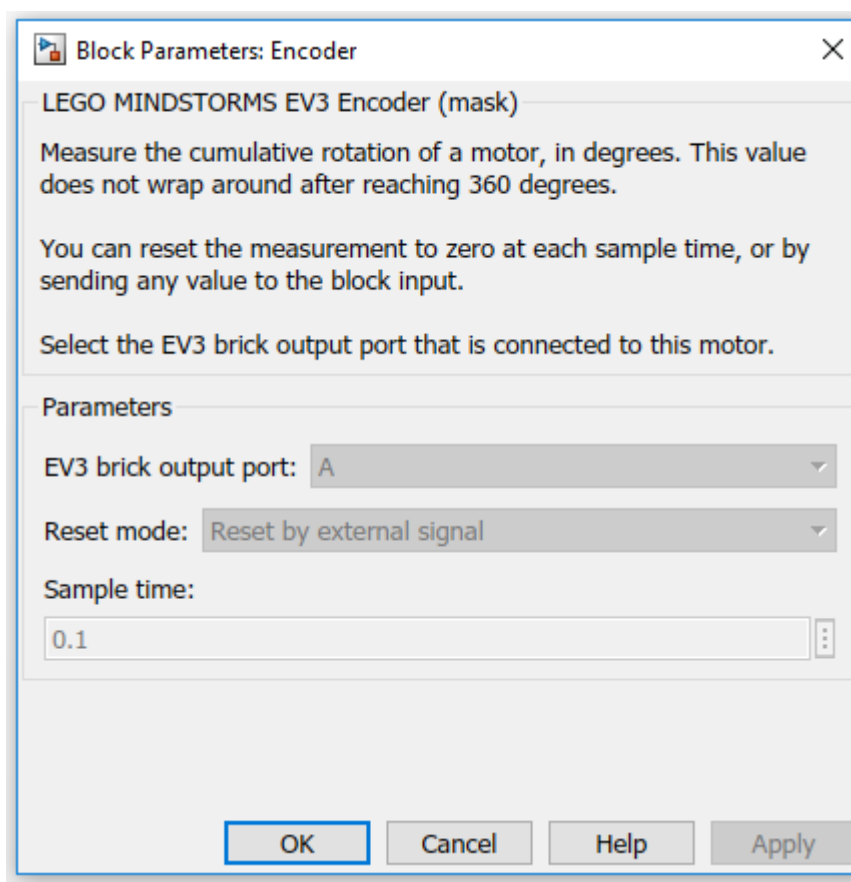


Ilustración 69. Bloque Encoder

Bloque Motor

Este bloque controla la potencia y la dirección de un motor EV3. Suele estar asociado a un bloque encoder que mida la rotación del motor. El único parámetro es el puerto al que se ha conectado en el ladrillo EV3. Los valores que aceptan están entre 100 y -100:

- 1 a 100: sentido positivo, de menos a más. Cuando la entrada es superior a 100, se satura en 100.
- -1 a -100: sentido negativo, de menos a más. Cuando la entrada es menor a -100, se satura en -100.

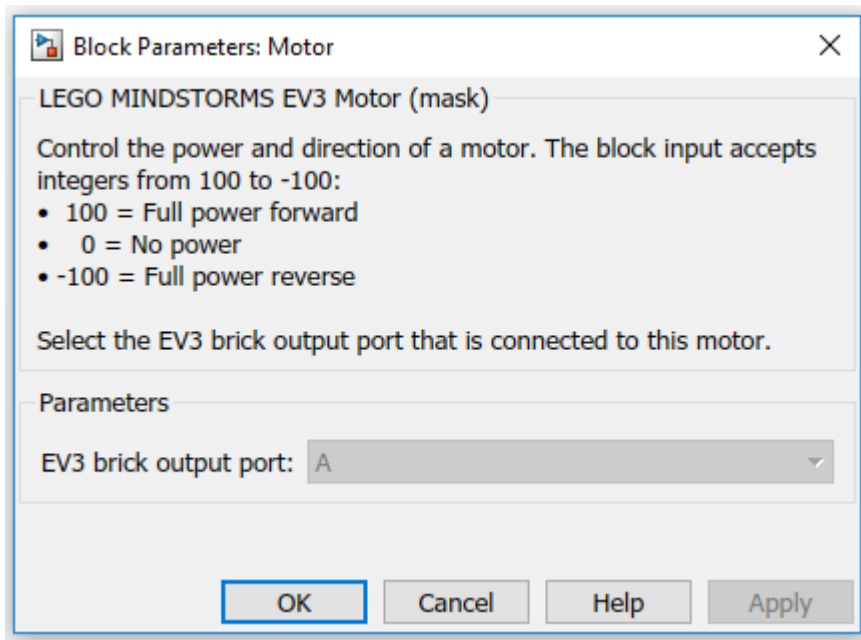


Ilustración 70. Bloque Motor

Bloque UDP Send

Este bloque se utiliza para enviar paquetes UDP a un dirección de red identificada por la dirección IP remota y lo parámetros del puerto IP remoto. En los parámetros se selecciona la dirección IP del dispositivo al que se quieren enviar los datos, el puerto remoto y por último el puerto local que por defecto se determina automáticamente.

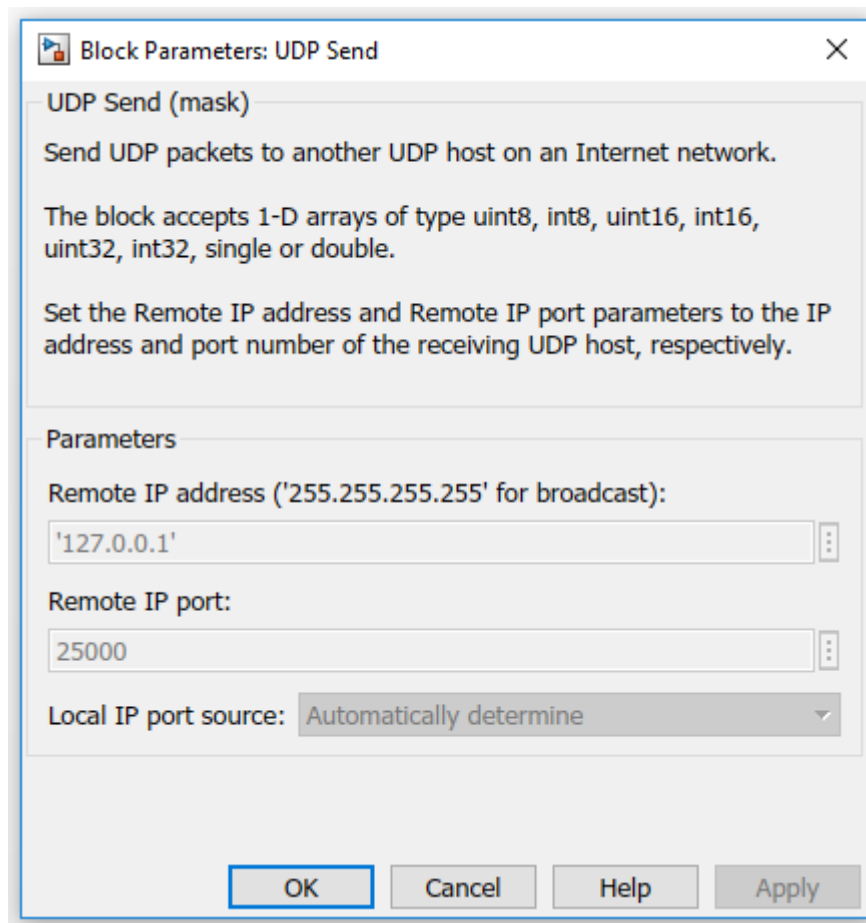


Ilustración 71. Bloque UDP Send

Bloque UDP Receive

Este bloque recibe paquetes UDP de la red local. Los parámetros de este bloque son: el puerto local a través del cual se reciben los datos (coincide con el puerto remoto usado por el bloque de envío utilizado), tipo de dato recibido, tamaño de los datos y frecuencia a la que se van a leer los datos.

Si se simula un modelo que contiene el bloque de recepción UDP en el ordenador (modo de simulación > normal) y se envían paquetes UDP a ese modelo desde un dispositivo, la salida de datos emite una cantidad desproporcionadamente grande de datos. Esto se debe a que el software Simulink simula el modelo como una aplicación de ejecución con un período más corto que la aplicación en tiempo real que se ejecuta en la placa. En otras palabras, la simulación de ejecución arroja los mismos datos porque está esperando que la aplicación en "tiempo más lento" envíe nuevos datos.

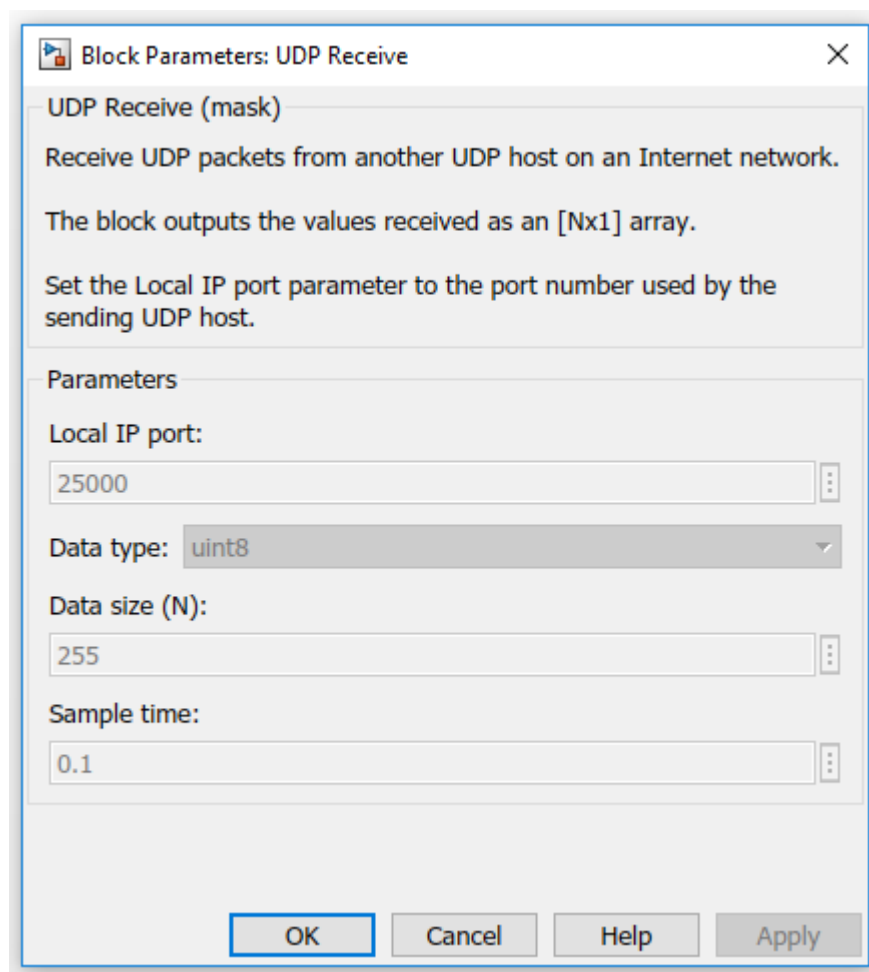


Ilustración 72. Bloque UDP Receive

3. Comunicación con LEGO MINDSTORMS EV3 Hardware

Una vez preparado el ladrillo EV3 para ejecutar cualquier modelo de *Simulink*, es necesario configurar el modelo que se va a cargar en el ladrillo. Para ello hay que seguir los siguientes pasos:

1. En el modelo de *Simulink*, pulsar **Tools > Run on Target Hardware > Prepare to Run...**
2. En la ventana **Configuration Parameters**, configurar la pestaña **Target Hardware** como **LEGO MINDSTORMS EV3**.
3. Por último en la pestaña **Host To Target Connection** se ponen los parámetros: ID del dispositivo y dirección IP el mismo.

Configurado ya el modelo, solo falta enviar dicho modelo al LEGO EV3, pulsando **Deploy To Hardware** en **Modo External**.

ANEXO 3. MOTOR DE CORRIENTE CONTINUA

En este anexo se va a detallar y explicar cómo se ha realizado la identificación y el cálculo de los parámetros del *Motor LEGO Power Functions XL* utilizado en la maqueta real del péndulo rotario invertido.

1. Modelo matemático de un motor de corriente continua

En primer lugar se desarrolla el modelo matemático que describe el comportamiento de un motor de corriente continua. El esquema de un motor CC puede observarse en la Ilustración 73.

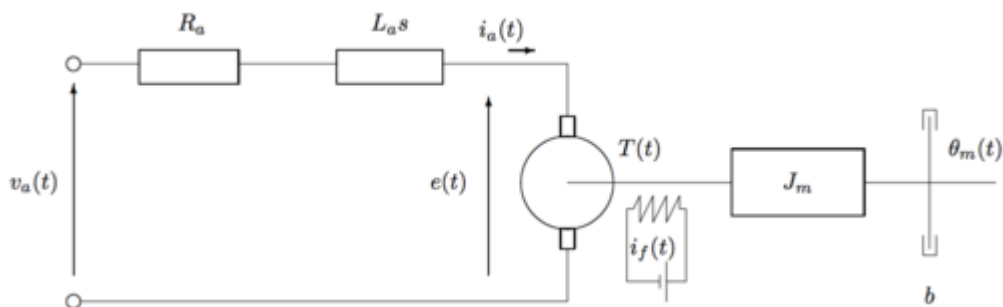


Ilustración 73. Esquema de un motor CC

Una de las partes más importantes del motor, el devanado del inducido, consiste en un arrollamiento de varias espiras que puede girar inmerso en un campo magnético constante. Dicho campo magnético puede ser generado por un imán permanente o por un devanado de excitación consistente en una bobina por la que circula una corriente de excitación $i_f(t)$, que se supone constante para que el campo sea también constante. Al circular una corriente $i_a(t)$ por el devanado del inducido, como resultado de la interacción con el campo magnético se ejerce sobre el un par $T(t)$ que es directamente proporcional al campo magnético y a la propia corriente del inducido $i_a(t)$. Dado que se supone el campo magnético constante, el par motor será proporcional a la corriente del inducido como se muestra en (1.1).

$$T(t) = K_t i_a(t) \quad (1.1)$$

ANEXOS

Por otra parte, el giro de las espiras del devanado del inducido en presencia del campo magnético, produce en bornes del mismo una caída de tensión o fuerza contra electromotriz $e(t)$, proporcional a su velocidad de giro, presente en la expresión (1.2).

$$e(t) = K_e \dot{\theta}_m(t) \quad (1.2)$$

Asimismo, el devanado del inducido es un conductor con una resistencia R_a y una inductancia L_a sobre el que, además, hay que considerar la fuerza contra electromotriz como una fuente de tensión dependiente de la velocidad de giro. La ecuación de la malla del inducido se presenta en (1.3).

$$v_a(t) = R_a i_a(t) + L_a \frac{di_a(t)}{dt} + K_e \dot{\theta}_m(t) \quad (1.3)$$

El par mecánico $T(t)$ desarrollado por el motor se emplea para imprimir una aceleración angular a la carga y en vencer la fuerza de fricción (la fricción viscosa es aproximadamente proporcional a la velocidad de giro).

$$T(t) = J_m \ddot{\theta}_m(t) + b \dot{\theta}_m(t) \quad (1.4)$$

Aplicando la transformada de Laplace a las ecuaciones (1.1), (1.2), (1.3) y (1.4), y teniendo en cuenta la expresión se obtienen (1.5), se obtienen (1.6), (1.7) y (1.8).

$$\Omega_m(s) = s\theta_m(s) \quad (1.5)$$

$$T(s) = K_t I_a(s) \quad (1.6)$$

$$V_a(s) = (R_a + sL_a)I_a(s) + K_e \Omega_m(s) \quad (1.7)$$

$$T(s) = (J_m s + b)\Omega_m(s) \quad (1.8)$$

ANEXOS

Agrupando términos, la función de transferencia que relaciona la tensión del inducido con la velocidad angular se observa en (1.9).

$$\frac{\Omega_m(s)}{V_m(s)} = \frac{K_t}{(J_ms + b)(L_as + R_a) + K_t K_e} \quad (1.9)$$

2. Obtención experimental de la planta

Para obtener experimentalmente la planta del motor real de corriente continua, se somete el Motor LEGO Power Functions XL a una señal para obtener su respuesta temporal, como se puede ver en la Ilustración 74. La señal que se usa para identificar el motor es un escalón, con esta señal se alimenta al motor y se adquieren los datos de la respuesta del mismo, para poder procesarlos y obtener una expresión numérica que relacione el valor PWM que se aplica al motor y la velocidad angular obtenida.

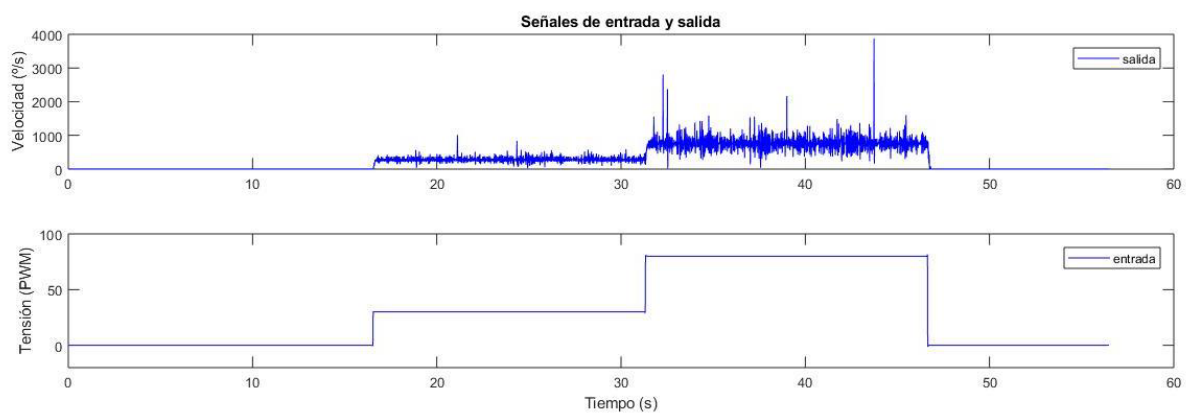


Ilustración 74. Gráfica de la respuesta del motor ante la señal de entrada

Utilizando la herramienta *Ident* de *Matlab*, utilizando los datos obtenidos en el experimento se llega a una planta aproximada que representa el comportamiento del sistema real (2.1). La planta obtenida queda reducida a una ganancia y un único polo, es decir, se trata de un sistema de primer orden.

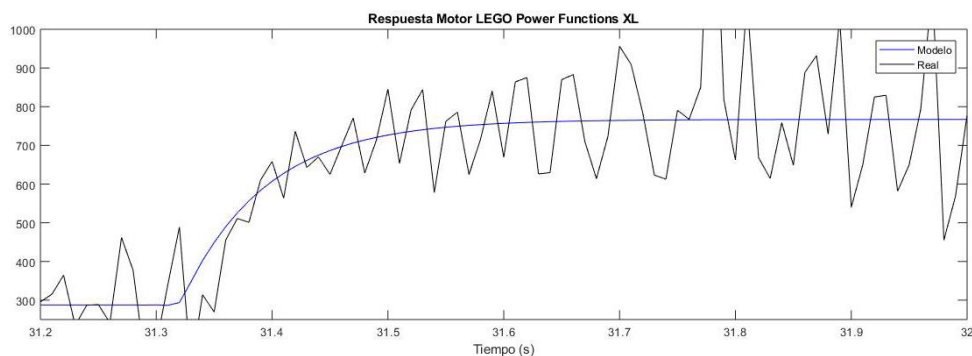


Ilustración 75. Gráfica del modelo del motor

$$\frac{\Omega_m(s)}{V_m(s)} = \frac{10}{0.1s + 1} \quad (2.1)$$

3. Características del motor

Utilizando los datos recogidos en la Ilustración 76 y en la Tabla 3, obtenidas de la [página web](#) recogida en la bibliografía, se van a obtener los parámetros característicos del motor. Para comprobar que estos datos son fiables, dicho experimento se comprobará con otro realizado sobre el motor de la maqueta.

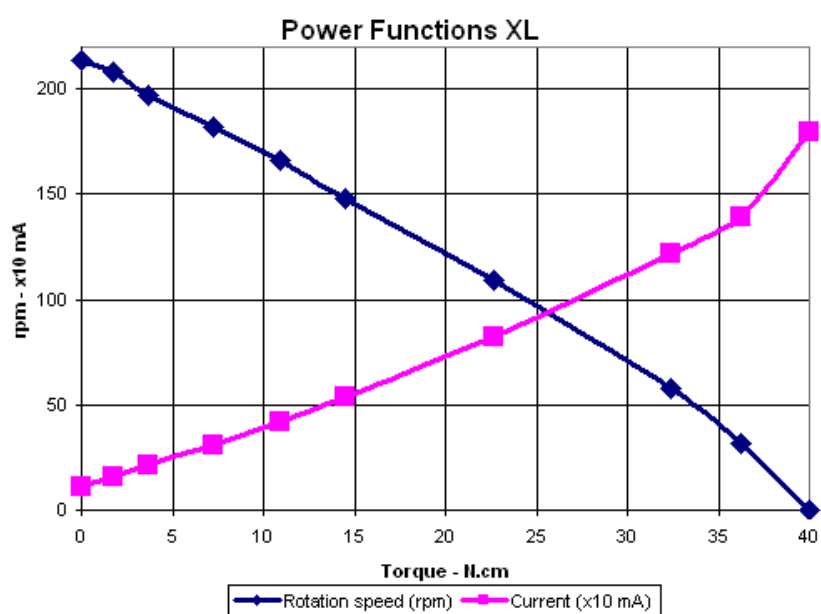


Ilustración 76. Gráfica Par-Velocidad-Corriente


 PF XL	Torque	Rotation speed	Current	Mechanical power	Electrical power	Efficiency
4.5 V	14.5 N.cm	43 rpm	0.52 A	0.65 W	2.34 W	28 %
7 V	14.5 N.cm	100 rpm	0.54 A	1.51 W	3.78 W	40 %
9 V	14.5 N.cm	146 rpm	0.55 A	2.21 W	4.95 W	45 %
12 V	14.5 N.cm	214 rpm	0.56 A	3.24 W	6.72 W	48 %

Tabla 3. Medidas experimento motor

ANEXOS

Teniendo en cuenta los datos recogidos en la Tabla 3 se pueden calcular las constantes eléctricas del motor: la resistencia del motor (R), la constante de par (K_t) y la constante de fuerza electromotriz (K_e).

Para obtener la constante de par:

El par máximo suministrado por el motor es de 14.5 N·cm cuando la corriente es 550 mA (tomando los datos aplicando al motor una tensión de 9V). Considerando que el par es proporcional a la corriente (3.1).

$$\tau(t) = K_t \cdot i(t) \quad (3.1)$$

Despejando K_t en (3.1).

$$K_t = \frac{\tau(t)}{i(t)} \quad (3.2)$$

Resolviendo (3.2).

$$K_t = \frac{0,145}{0,55} = 0,26 \text{ N} \cdot \text{cm/A} \quad (3.3)$$

Se considera que en un motor de corriente continua la constante de par (K_t) y la constante de fuerza electromotriz (K_e) son idénticas en el sistema internacional de unidades (SI), por lo tanto:

$$K_t = 0,26 \text{ N} \cdot \text{cm/A} \quad (3.4)$$

$$K_e = 0,26 \text{ V/(rad/s)} \quad (3.5)$$

ANEXOS

Para obtener la resistencia del motor, se realizan los siguientes cálculos (considerando los valores a una tensión de 9V):

Conociendo las ecuaciones (3.6) y (3.7), se obtiene (3.8).

$$P = V \cdot I \quad (3.6)$$

$$V = V_0 + R \cdot I \quad (3.7)$$

$$P = V_0 \cdot I + R \cdot I^2 \quad (3.8)$$

Conociendo la potencia mecánica es posible obtener V_0 .

$$W = P \cdot \varepsilon$$

$$W = 4,95 \cdot 0,45 = 2,2275 \quad (3.9)$$

$$W = V_0 \cdot I$$

$$V_0 = \frac{2,2275}{0,55} = 4,05 \quad (3.10)$$

Resolviendo (3.8) se obtiene que la resistencia del motor es de 9Ω .

Una vez conocidas las constantes del motor LEGO Power Functions XL, se realiza un experimento sobre él mismo alimentándolo con una fuente de alimentación. Se mide la velocidad que suministra el motor y la intensidad que circula por él. Con esos datos y utilizando la ecuación (1.3), despreciando el factor inductivo por ser muy pequeño, es posible hallar la constante de fuerza contraelectromotriz (K_e) que debería coincidir con la obtenida con los anteriores cálculos. Además suponiendo que el par eléctrico coincide con el par mecánico en régimen permanente es posible obtener la fricción viscosa del motor (b), que tomando el valor medio se considera $0,0022 \text{ Nm/(V/rad/s)}$.

Tensión (V)	Intensidad (mA)	Velocidad (rpm)	K_e (V/(rad/s))	b (Nm/(rad/s))
0	0	0	0,00	0
1	45	25	0,23	0,0039
2	60	50	0,28	0,0032
3	70	75	0,30	0,0027
4	75	100	0,32	0,0023
5	85	125	0,32	0,0021
6	90	150	0,33	0,0019
7	100	175	0,33	0,0018
8	105	200	0,34	0,0017
9	110	225	0,34	0,0016

Tabla 4. Datos obtenidos del experimento

ANEXOS

Comparando las velocidades del experimento a partir del cual se han obtenido las constantes eléctricas del motor con éste, es posible deducir que el motor se comporta de la misma forma en ambos experimentos. Dicho comportamiento se muestra en la Ilustración 77 y en la Ilustración 78, que representan la velocidad frente a la tensión entre bornes del motor en el experimento de la página web y en el propio, respectivamente.

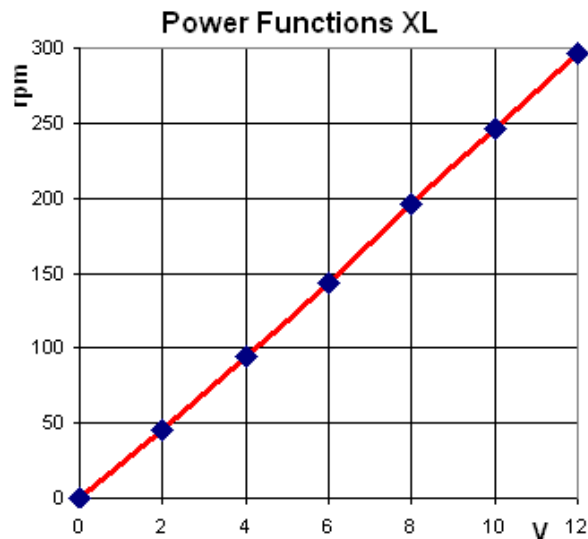


Ilustración 77. Gráfica Tensión-Velocidad

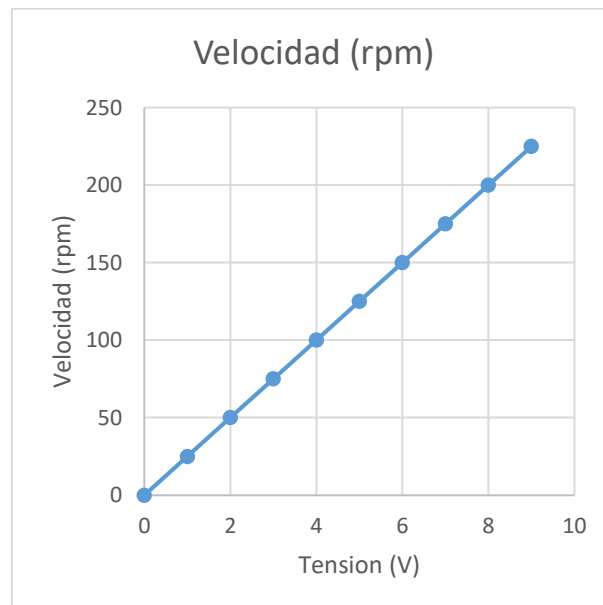


Ilustración 78. Gráfica Tensión-Velocidad

ANEXOS

Por último, utilizando la ecuación (1.9), despreciando el efecto inductivo por ser muy pequeño, y la ecuación (2.1), es posible obtener el momento de inercia del motor J_m .

En primer lugar es necesario transformar la ecuación (1.9), para que tenga la misma forma que la (2.1), para así poder comparar la constante de tiempo y poder obtener la inercia del motor. La ganancia de la ecuación (2.1) corresponde a la función de transferencia en unidades de PWM y grados por segundo, por lo que no coincidirá con la ecuación la ganancia de (3.11). En el caso de la constante de tiempo el valor de la ecuación (2.1) si será válido, y ese es valor necesario para el cálculo del momento de inercia. Resolviendo la ecuación (3.12) con los valores de los parámetros del motor se obtiene un momento de inercia (J_m) de $9.7 \cdot 10^{-4} \text{ kg} \cdot \text{m}^2$.

$$\frac{\Omega_m(s)}{V_m(s)} = \frac{\frac{K_t}{R_a b + K_t K_e}}{\frac{R_a J_m}{R_a b + K_t K_e} s + 1} \quad (3.11)$$

$$\frac{R_a J_m}{R_a b + K_t K_e} = 0.1 \quad (3.12)$$

“CONTROL DE UN PÉNDULO INVERTIDO ROTATORIO CON HARDWARE DE BAJO COSTE”



DOCUMENTO N° 4 PLIEGO DE CONDICIONES

Peticionario:	Universidad de La Rioja
Informantes:	Fernando Chicote Navaridas
	Estudiante de Ingeniería Electrónica Industrial y Automática

ÍNDICE

PLIEGO DE CONDICIONES.....	116
1. Introducción al pliego de condiciones	116
2. Condiciones Generales	116
3. Condiciones Administrativas	117
4. Normativa y Reglamentación	118
Reglamento relacionado a productos electrónicos.....	118
Normativa relacionada con materiales y equipos	118
5. Condiciones facultativas.....	119
Dirección	119
Libro de órdenes	119
Modificaciones	119
6. Condiciones de materiales y equipos.....	120
Condiciones técnicas de los materiales.....	120
7. Condiciones económicas	121
Errores en el proyecto.....	121
Liquidación.....	121
8. Disposición final.....	121

PLIEGO DE CONDICIONES

1. Introducción al pliego de condiciones

El autor de este proyecto ha cursado los estudios de Grado en Ingeniería Electrónica Industrial y Automática en la Universidad de La Rioja, cumpliendo con la normativa establecida por la Escuela Superior de Ingeniería Industrial en la normativa de trabajo fin de grado.

El objeto de este Pliego de Condiciones es recoger y establecer todas las disposiciones técnicas, administrativas y económicas, y las normativas que ha de regir este proyecto.

El diseño de este proyecto y sus características han sido descritos en detalle en la memoria del proyecto y sus anexos.

Las condiciones que se especifican en este documento tratan de cumplir con la calidad esperada para este proyecto. En caso de no realizarse según estas condiciones, el proyectista no se responsabilizará de los fallos o averías que puedan ocasionarse en su funcionamiento, y los problemas derivados repercutirían sobre terceras personas.

Todas las modificaciones que puedan sufrir el proyecto y sus documentos deberán ser aprobadas por el ingeniero o proyectista.

2. Condiciones Generales

Este proyecto se ajusta a los reglamentos y normativas electrónicas vigentes. Una vez terminado el proyecto se podrán llevar a cabo modificaciones pero siempre bajo la supervisión del proyectista.

La propiedad intelectual del autor y director del Trabajo Fin de Grado se regirá por el Real Decreto Legislativo 1/1996, de 12 de abril, por el que se aprueba el texto refundido de la Ley de Propiedad Intelectual.

3. Condiciones Administrativas

El proyecto constará de los siguientes documentos:

- Un Índice General que indicará la página de comienzo de cada uno de los documentos que forman el proyecto.
- Una Memoria donde se considerarán las necesidades a satisfacer y los factores técnicos a tener en cuenta entrando en profundidad en las posibles soluciones técnicas y en la justificación de la solución elegida.
- Anexos donde se recogerá la documentación considerada de interés para ampliar la descripción detallada de los componentes del sistema.
- Pliego de Condiciones, este documento en el que se establecen las diferentes condiciones técnicas, económicas y administrativas para que proyecto pueda materializarse, evitando posibles malinterpretaciones.
- Presupuesto donde se recogerá el coste de todos los componentes utilizados y la suma total que, junto a la mano de obra, dará el coste final del proyecto. Dicho presupuesto contiene la valoración económica global, desglosada y ordenada por partidas.

4. Normativa y Reglamentación

El proyecto estará regido tanto por la normativa española como por la internacional:

Reglamento relacionado a productos electrónicos

Respecto al desarrollo de productos electrónicos, se pueden encontrar en AENOR (Asociación Española de Normalización y Certificación) las siguientes normativas:

- Norma UNE1302—2:1973 Vocabulario electrotécnico. Electrónica.
- Norma UNE-EN611000-4-3-1998. Compatibilidad electromagnética.

Este proyecto debido a sus características se rige bajo el reglamento de Baja Tensión:

“Se calificará como instalación eléctrica de baja tensión todo conjunto los aparatos y de circuitos asociados en previsión de un fin particular: producción, conversión, transformación, transmisión, distribución o utilización de la energía eléctrica, cuyas tensiones nominales sean iguales o inferiores a 1000 V para corriente alterna y 1500 V para corriente continua.”

Normativa relacionada con materiales y equipos

Los materiales y equipos de este proyecto deben cumplir los estándares nacionales e internacionales en vigor y obligado cumplimiento. Entre otros:

- UNE 20-324 Grados de protección de los envolventes del material eléctrico de baja tensión.
- UNE 20-334 Conductos para instalaciones eléctricas.
- UNE 21-401 Conductores eléctricos aislados.
- UNE 21-402 Conductores eléctricos aislados y desnudos.

5. Condiciones facultativas

Dirección

La dirección del montaje del será llevada a cabo, en su totalidad por el ingeniero proyectista, o por cualquier otra persona en la que éste delegue, atendiendo a la capacidad de dicha persona para responsabilizarse de dicha dirección.

Una vez realizada la instalación, ésta podrá ser utilizada por cualquier persona con conocimientos demostrables suficientes sobre el sistema y sus componentes, las tecnologías implicadas y el funcionamiento global y, por partes, del sistema.

En caso de avería o pérdida de información por una utilización incorrecta, el ingeniero proyectista o la persona en la que haya delegado la dirección del proyecto, quedan exentas de responsabilidad.

Libro de órdenes

El montaje e instalación de todos los elementos que componen el proyecto se realizará atendiendo al siguiente orden de prioridad en caso de que haya alguna contradicción:

- Presupuesto.
- Pliego de condiciones
- Memoria

Este libro de órdenes debe estar conforme al Decreto 462/1971 de 11 de marzo, y la Orden de 9 de junio de 1971.

Modificaciones

Si fuera necesario realizar alguna modificación en el presente proyecto, deberá comunicarse con anterioridad a su realización al Ingeniero Director, quién deberá dar la correspondiente autorización.

En caso de realizarse modificaciones en la instalación que no hayan sido previamente comunicadas y autorizadas por el ingeniero Director, las consecuencias que dichos cambios puedan ocasionar serán de total responsabilidad del instalador que las realice.

Respecto a los cambios en la instalación realizados por el propietario de la misma, no serán tratados de forma especial y, en ningún caso, quedan eximidos de la autorización del ingeniero Director.

6. Condiciones de materiales y equipos

A continuación se detallan las condiciones tanto de hardware como de software con las que hay que contar para hacer uso de la aplicación del proyecto.

Condiciones técnicas de los materiales

Todos los materiales y componentes, utilizados en el proyecto, deben cumplir todas las especificaciones técnicas que aparecen descritas en la Memoria y deben cumplir asimismo todas las normas descritas en este pliego de condiciones.

Si se considera necesario reemplazar algún componente o material por otro, los nuevos deberán tener las mismas características que los reemplazados, inhibiéndose el ingeniero proyectista de cualquier responsabilidad por fallo, si no se cumplen estos requisitos.

Para el desarrollo e implementación del proyecto se deberá disponer de un PC que será el encargado de recoger los datos, analizarlos y procesarlos. Para ello deberá disponer al menos del siguiente software:

- Windows 10
- Matlab – Simulink R2017b

7. Condiciones económicas

Errores en el proyecto

En el caso de existir algún tipo de error en el proyecto se avisará inmediatamente al proyectista y se le informará con detalle de los errores encontrados.

Además se dejará de usar la aplicación hasta que los errores queden solucionados para prevenir cualquier tipo de daño.

Liquidación

Terminada la instalación del péndulo, se procederá a la liquidación final, en la que se incluye el importe de las unidades de realización, así como las posibles modificaciones del proyecto que hayan sido aprobadas por la dirección del proyecto.

Al suscribir el contrato, el cliente habrá de abonar el 80% del presupuesto. El 20% quedará como garantía durante los seis primeros meses, a partir de la fecha de puesta en marcha del sistema.

Si transcurridos seis meses desde la puesta en marcha no se ha manifestado ningún defecto o error de funcionamiento, el cliente abonará el 20% que estaba pendiente. A partir de ese momento, se considerarán concluidos los compromisos entre ambas partes, a excepción del periodo de garantía.

8. Disposición final

Las partes contratantes, tanto la dirección del proyecto como la empresa cliente, se ratifican en el contenido del presente pliego de condiciones, que tiene igual validez, a todos los efectos, que una escritura pública, prometiendo su fiel cumplimiento.

“CONTROL DE UN PÉNDULO INVERTIDO ROTATORIO CON HARDWARE DE BAJO COSTE”



DOCUMENTO N° 5 MEDICIONES

Peticionario:	Universidad de La Rioja
Informantes:	Fernando Chicote Navaridas
	Estudiante de Ingeniería Electrónica Industrial y Automática

ÍNDICE

MEDICIONES	124
-------------------------	------------



MEDICIONES

C01: Materiales para la construcción de la maqueta		
C01.1	Kit LEGO MINDSTORMS Education EV3	1
C01.2	Set de expansión EV3	1
C01.3	WNA1100: Adaptador USB inalámbrico	1
C01.4	Motor Lego PF XL	1
C01.5	GlideWheel-M Encoders	2
C02: Mano de obra		
C02.1	Horas de investigación	100
C02.2	Horas de diseño	140
C02.3	Horas de montaje	10
C02.4	Horas de programación	50

“CONTROL DE UN PÉNDULO INVERTIDO ROTATORIO CON HARDWARE DE BAJO COSTE”



DOCUMENTO N° 6 PRESUPUESTO

Peticionario:	Universidad de La Rioja
Informantes:	Fernando Chicote Navaridas
	Estudiante de Ingeniería Electrónica Industrial y Automática

ÍNDICE

PRESUPUESTO.....	127
1. Cuadro de precios	127
2. Presupuesto	128
3. Resumen del presupuesto	129

PRESUPUESTO

1. Cuadro de precios

C01: Materiales para la construcción de la maqueta		
C01.1	Kit LEGO MINDSTORMS Education EV3	447.69
C01.2	Set de expansión EV3	131.88
C01.3	WNA1100: Adaptador USB inalámbrico	10
C01.4	Motor Lego PF XL	11.99
C01.5	GlideWheel-M Encoders	57
C02: Mano de obra		
C02.1	Horas de investigación	40
C02.2	Horas de diseño	35
C02.3	Horas de montaje	30
C02.4	Horas de programación	35

2. Presupuesto

CÓDIGO	DESCRIPCIÓN	CANTIDAD	PRECIO	IMPORTE
C01: Materiales para la construcción de la maqueta				
C01.1	Kit LEGO MINDSTORMS Education EV3	1	447.69	447.69
C01.2	Set de expansión EV3	1	131.88	131.88
C01.3	WNA1100: Adaptador USB inalámbrico	1	10	10
C01.4	Motor Lego PF XL	1	11.99	11.99
C01.5	GlideWheel-M Encoders	2	57	114
Total capítulo				715.56

CÓDIGO	DESCRIPCIÓN	CANTIDAD	PRECIO	IMPORTE
C02: Mano de obra				
C02.1	Horas de investigación	100	40	4000
C02.2	Horas de diseño	140	35	4900
C02.3	Horas de montaje	10	30	300
C02.4	Horas de programación	50	35	1750
Total capítulo				10950

PRESUPUESTO

3. Resumen del presupuesto

CAPÍTULO RESUMEN				EUROS	%
C01	Materiales para la construcción de la maqueta			715.56	6.13
C02	Mano de obra			10950	93.87

TOTAL EJECUCIÓN MATERIAL 11665.56

13% Gastos generales 1516.523

6% Beneficio industrial 699.934

SUMA DE GG y BI 2216.456

21% de IVA 2915.223

TOTAL PRESUPUESTO GENERAL 16797.24

Asciende el presupuesto general a la expresada cantidad de DIECISEIS MIL SETECIENTOS NOVENTA Y SIETE EUROS con VEINTICUATRO CÉNTIMOS.

FIRMADO:

D. Fernando Chicote Navaridas

Logroño a 5 de marzo de 2018